

METHOD FOR SUPPORTING CONSTITUTION GENERATION OF APPLICATION PROGRAM

Publication number: JP9258975

Publication date: 1997-10-03

Inventor: FUTAKI SEIJI; YUURA KATSUHIKO

Applicant: HITACHI LTD

Classification:


- international: G06F9/06; G06F9/44; G06F9/445; G06F9/06; G06F9/44; G06F9/445; (IPC1-7): G06F9/06

- European: G06F9/44G4M; G06F9/445L

Application number: JP19960067931 19960325

Priority number(s): JP19960067931 19960325

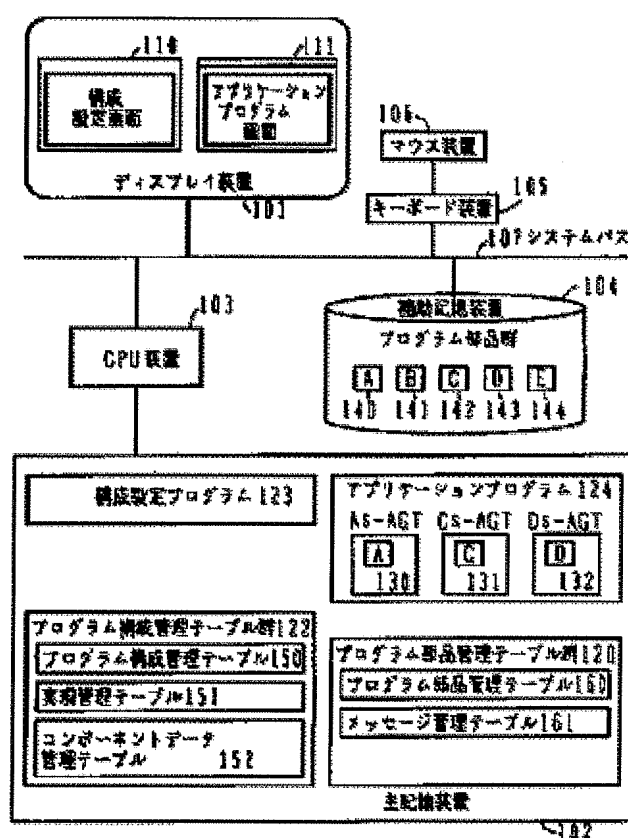
Also published as:

 US6072949 (A1)

Report a data error here

Abstract of JP9258975

PROBLEM TO BE SOLVED: To provide the constitution generation support method of an application program, which sequentially executes and evaluates combination in a process for combining parts. **SOLUTION:** The system is provided with a program parts management table 120, a program constitution management table 122, a program parts entry storing the parts entry of the program parts management table 120 and a message distribution program for delivering a message call from outside to one of the program parts in a component, which is obtained from the program parts entry. The parts entry is set in the program parts entry of component data in accordance with indication input containing the designation of the program parts. The application program is executed by exchanging a message between component data, and the message delivered by individual component data are processed by the message delivery program.



Data supplied from the esp@cenet database - Worldwide

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平9-258975

(43)公開日 平成9年(1997)10月3日

(51)Int.Cl.⁶

G 0 6 F 9/06

識別記号

5 3 0

庁内整理番号

F I

G 0 6 F 9/06

技術表示箇所

5 3 0 W

審査請求 未請求 請求項の数8 O L (全 43 頁)

(21)出願番号

特願平8-67931

(22)出願日

平成8年(1996)3月25日

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 二木 誠司

神奈川県川崎市幸区鹿島田890番地の12

株式会社日立製作所情報通信開発本部内

(72)発明者 湯浦 克彦

神奈川県川崎市幸区鹿島田890番地の12

株式会社日立製作所情報通信開発本部内

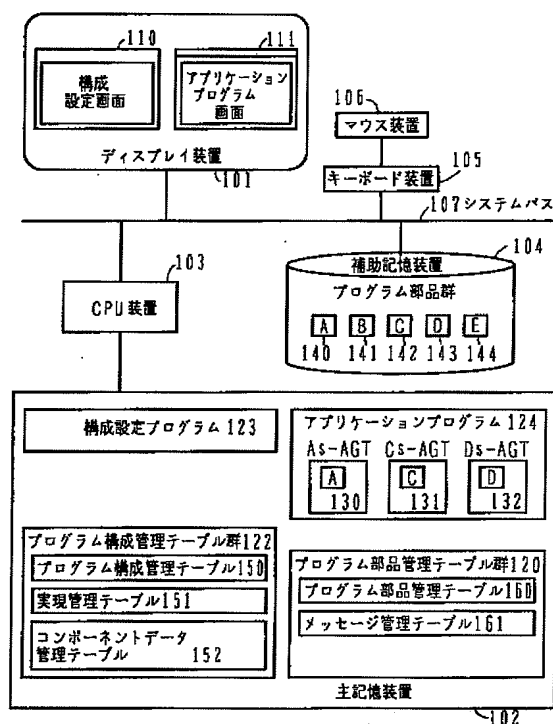
(74)代理人 弁理士 春日 譲

(54)【発明の名称】 アプリケーションプログラムの構成作成支援方法

(57)【要約】

【課題】本発明の目的は、部品を組み合わせていく過程で、次々に実行し、評価を行うことを可能とするアプリケーションプログラムの構成作成支援方法を提供するにある。

【解決手段】プログラム部品管理テーブル120と、プログラム構成管理テーブル122と、プログラム部品管理テーブル120の部品エントリを格納するプログラム部品エントリ910と、外部からのメッセージ呼び出しを、プログラム部品エントリ910から取得するコンポーネント内のプログラム部品の1つに配送するメッセージ配送プログラム900とを管理するとともに、構成レコードに1対1に対応するコンポーネントデータ130とを備えている。プログラム部品の指定を含む指示入力に応じ、コンポーネントデータのプログラム部品エントリ910に、この部品エントリを設定し、アプリケーションプログラムの実行は、コンポーネントデータ間のメッセージ交換によって行い、個々のコンポーネントデータに配送されたメッセージは、メッセージ配送プログラムによって処理する。



【特許請求の範囲】

【請求項1】 計算機システムの実行を制御するアプリケーションプログラムを機能単位によって分割した複数のプログラム部品を組み合わせて、アプリケーションプログラムを構成作成することを支援するアプリケーションプログラムの構成作成支援方法において、個々の上記プログラム部品を判別可能とする識別子と、上記プログラム部品を機能によって分類する機能分類と、上記プログラム部品のメッセージインタフェースと、実行オブジェクトとを含む部品レコードから構成されるプログラム部品管理テーブルと、

作成対象となるアプリケーションプログラムの構成を、上記プログラム部品管理テーブルに格納された上記プログラム部品の識別子とプログラム名称とを含む構成レコードによって記述するプログラム構成管理テーブルと、上記プログラム部品管理テーブルの部品エントリを格納するプログラム部品エントリと、外部からのメッセージ呼び出しを、上記プログラム部品エントリから取得するコンポーネント内のプログラム部品の1つに配送するメッセージ配送プログラムとを管理するとともに、上記構成レコードに1対1に対応するコンポーネントデータとを備え、

(A) 上記アプリケーションプログラムの個々の機能を実現する上記プログラム部品に関し、上記識別子による上記プログラム部品の指定を含む指示入力に応じ、(A-1) この機能に対応した上記コンポーネントデータを取得し、(A-2) 上記プログラム部品管理テーブルから上記識別子を含む上記部品レコードを取得し、この部品レコードを参照して、上記コンポーネントデータの上記プログラム部品エントリに、この部品エントリを設定し、

(B) 上記アプリケーションプログラムの実行の指示入力に応じ、(B-1) 上記プログラム構成管理テーブルの上記構成レコードに対応した全てのコンポーネントデータを含んだ実行オブジェクトを作成し、(B-2) この実行オブジェクトの実行は、上記コンポーネントデータ間のメッセージ交換によって行い、(B-3) 個々のコンポーネントデータに配送されたメッセージは、上記メッセージ配送プログラムによって処理することを特徴とするアプリケーションプログラムの構成作成支援方法。

【請求項2】 請求項1記載のアプリケーションプログラムの構成作成支援方法において、上記コンポーネントデータは、プログラム部品のリストを格納する候補リストを有し、この候補リストには、上記プログラム部品エントリとの交換を意図したプログラム部品管理テーブルのプログラム部品を格納し、

上記候補リスト内に格納されたプログラム部品のうち、指示された上記候補リスト内の上記プログラム部品を上

記プログラム部品エントリに設定することを特徴とするアプリケーションプログラムの構成作成支援方法。

【請求項3】 請求項2記載のアプリケーションプログラムの構成作成支援方法において、上記ステップ(A)におけるプログラム部品の指示入力において、識別子としてジョーカー部品が指示された際には、

上記ステップ(B-3)において、上記メッセージ配送プログラムは、配送されたメッセージの形式に応じ、

上記コンポーネントデータ内の候補リストに格納された個々の部品レコードについて、この部品レコードから取得したメッセージ形式と、配送されたメッセージとを比較し、一致する部品レコードから取得した実行オブジェクトにメッセージを配送することを特徴とするアプリケーションプログラムの構成作成支援方法。

【請求項4】 請求項1記載のアプリケーションプログラムの構成作成支援方法において、

上記アプリケーションプログラムの実行が指示された際に検索される上記コンポーネントデータの候補リスト内の部品レコードの検索順序を変更可能とすることを特徴とするアプリケーションプログラムの構成作成支援方法。

【請求項5】 請求項1記載のアプリケーションプログラムの構成作成支援方法において、

上記アプリケーションプログラムの1つの部品組み合わせに対する名称として、実現名称を与え、

上記プログラム構成管理テーブルの構成レコードに、上記実現名称を格納する実現名エントリを設け、

上記コンポーネントデータ内に、上記部品レコードと上記実現名称の組み合わせを要素とする履歴リストを設け、

アプリケーションプログラムの構成の保存が、上記実現名称を付与して指示された場合には、

上記プログラム構成管理テーブルの構成レコードの上記実現名称エントリに、この実現名称を記録し、

上記コンポーネントデータ内の上記履歴管理リストに、上記プログラム部品エントリに格納された部品レコードと実現名称を記録することを特徴とするアプリケーションプログラムの構成作成支援方法。

【請求項6】 請求項5記載のアプリケーションプログラムの構成作成支援方法において、

上記実現名称を表示する複数のノードを、実現の依存関係によってアーク結線した形式によってディスプレイ装置上に表示することを特徴とするアプリケーションプログラムの構成作成支援方法。

【請求項7】 請求項1記載のアプリケーションプログラムの構成作成支援方法において、

上記コンポーネントデータ内に、特定の呼び出しインタフェース情報とその実行オブジェクトを格納するプログラム例エントリを設け、

上記メッセージ配送プログラムは、外部からのメッセージ呼び出し形式を参照し、上記プログラム例エントリのインタフェース情報と一致するなら、上記実行オブジェクトにメッセージ配送することを特徴とするアプリケーションプログラムの構成作成支援方法。

【請求項8】 請求項1記載のアプリケーションプログラムの構成作成支援方法において、

上記プログラム部品管理テーブルの部品レコードに、接続実績リストを格納するエントリを設け、

1つのアプリケーションプログラムが保存された場合、
上記プログラム構成管理内テーブルで同じアプリケーションプログラムに含まれたものを、各部品レコードの上
記接続実績リストに格納することを特徴とするアプリケーションプログラムの構成作成支援方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、アプリケーションプログラムの構成作成支援方法に係り、特に、作成済みであり、機能分割されたプログラム部品を利用し、新たなプログラムの構成要素として利用することを支援する
アプリケーションプログラムの構成作成支援方法に関する。

【0002】

【従来の技術】アプリケーションプログラムを作成する場合、機能ごとに分割されたプログラム部品を使用することが多くなっている。販売されるパッケージプログラムの中にも、その一部が部品化されて他のプログラムから呼び出して利用できるものもある。

【0003】例えば、マイクロソフトジャーナル日本語版、No. 37 (1995年7月号), p. 53-65
に記載されているように、マイクロソフト社のデータベース管理プログラムの「Access」では、データを管理するテーブル部品を中心とし、統計解析プログラム、グラフプログラムなどのプログラム部品を結合可能な構成となっている。また外部のプログラムから、こういったプログラム部品を呼び出すことも可能である。プログラム部品としては、この他に、入出力画面、表計算のワークシート、データ分析、数値計算などが流通している。

【0004】ここで、部品を利用した新しいプログラムとする作業としては、部品の機能を改造または追加する作業と、部品間を接続するプログラムを記述する作業がある。すなわち、アプリケーションプログラムの作成作業は、プログラム部品を作成、変更する作業と、部品を組み合わせる作業から構成されている。

【0005】プログラム部品を使用することで、その部分をプログラマーが作成しなくても良いことになるため、プログラムのミスを防止し、また、検査の手間を省くことができるといった効果がある。この結果として、アプリケーションプログラムの生産性向上が期待され

る。

【0006】部品を活用したプログラミングにおいては、適切なプログラム部品を選択することが重要となる。上述した「Access」の例を含め、多数のプログラム部品が提供されているが、この中から機能、性能やプログラムサイズなどすべての要求を満たす最適な部品を見つけることは容易なことではない。

【0007】最近、アプリケーションプログラムの構築支援環境は、リポジトリ技術として研究開発されてきている。即ち、プログラム部品の選択を支援する方法として、データベースの利用が考えられる。例えば、プログラム部品に、機能や性能などの情報を付加してデータベース化する。そして、例えば、「株価の推移チャートを表示する」機能を持ち、かつ「チャート表示速度が0.1秒以内」の性能を持つプログラム部品を検索することが可能となる。プログラマーは、検索結果のプログラム部品をデータベースから取得し、これをアプリケーションプログラムに組み込むこととなる。

【0008】

【発明が解決しようとする課題】プログラム部品としては、現在、膨大なプログラム部品が流通しており、作成しようとするアプリケーションプログラムにおいて使用可能なものもいくつも存在する。単純に考えて、10個に機能分割されるアプリケーションプログラムの機能毎に、それぞれ10のプログラム部品候補があれば、100の組み合わせを検証する必要がある。従来は、これらのうち、開発者が有望であると考えいくつかの組み合わせのみを検証していた。ただし、検証手順は自動化されておらず、膨大な手間が必要であった。組み合わせに応じてプログラム部品の書換えが生じる場合など、全体のうちの一部分が不十分である場合に全体の評価手順を狂わせる要因となる。

【0009】また、リポジトリを検索した結果であるプログラム部品が、必ずしも開発中のプログラム構成に最適なものであるとは限らない。プログラム部品は、さまざまな状況で使用されるため、常に同じ性能を発揮するとは限らず、また、他に使用されるプログラム部品によって全体の性能が大きく変化するためである。

【0010】また、プログラムの評価尺度としては、機能や速度、プログラムサイズの他にも、保守性、信頼性、値段、インタフェースプログラムの作成環境などが存在する。したがって、従来は、プログラム部品を組み立ててアプリケーションプログラムを作成した最終段階になって、はじめて性能が未達成であることが判明することもあるという問題があった。

【0011】なお、例えば、コンピュータ・サイエンス誌bit別冊、「Common Lisp オブジェクトシステム(CLOS)」(1989年1月発行)、共立出版社発行、p. 27-57に記載されているように、支援における視点を変えて、実際に部品を動作させ

て、それが適切であるかどうかを確認する方法も開発されている。CLOSの処理系では、プログラムの実行の一時中断中に、すでに実行された関数を変更することが可能である。さらに、プログラム部品とも解釈されるオブジェクトの定義を変更することも可能である。また、複数のオブジェクトに同じメッセージインタフェースを備えさせることで、動的に（実行時環境に依存して）メッセージを受け取るオブジェクトを変更することも可能である。かかるCLOSの処理系においては、実行中に部品を組み替えることが可能となっているが、取り替えは自動化されておらず、どのような組み合わせとするかはプログラムによって指定する必要がある。また、本発明の課題において、実行中に別のプログラム部品と変更することは必要ではない。

【0012】本発明の目的は、部品を組み合わせしていく過程で、次々に実行し、評価を行うことを可能とするアプリケーションプログラムの構成作成支援方法を提供するにある。

【0013】

【課題を解決するための手段】上記目的を達成するために、本発明は、計算機システムの実行を制御するアプリケーションプログラムを機能単位によって分割した複数のプログラム部品を組み合わせ、アプリケーションプログラムを構成作成することを支援するアプリケーションプログラムの構成作成支援方法において、個々の上記プログラム部品を判別可能とする識別子と、上記プログラム部品を機能によって分類する機能分類と、上記プログラム部品のメッセージインタフェースと、実行オブジェクトとを含む部品レコードから構成されるプログラム部品管理テーブル(群)と、作成対象となるアプリケーションプログラムの構成を、上記プログラム部品管理テーブルに格納された上記プログラム部品の識別子とプログラム名称とを含む構成レコードによって記述するプログラム構成管理テーブル(群)と、上記プログラム部品管理テーブル(群)の部品エントリを格納するプログラム部品エントリと、外部からのメッセージ呼び出しを、上記プログラム部品エントリから取得するコンポーネント内のプログラム部品の1つに配送するメッセージ配送プログラムとを管理するとともに、上記構成レコードに1対1に対応するコンポーネントデータとを備え、
(A) 上記アプリケーションプログラムの個々の機能を実現する上記プログラム部品に関し、上記識別子による上記プログラム部品の指定を含む指示入力に応じ、(A-1) この機能に対応した上記コンポーネントデータを取得し、(A-2) 上記プログラム部品管理テーブル(群)から上記識別子を含む上記部品レコードを取得し、この部品レコードを参照して、上記コンポーネントデータの上記プログラム部品エントリに、この部品エントリを設定し、
(B) 上記アプリケーションプログラムの実行の指示入

力に応じ、(B-1) 上記プログラム構成管理テーブル(群)の上記構成レコードに対応した全てのコンポーネントデータを含んだ実行オブジェクトを作成し、(B-2) この実行オブジェクトの実行は、上記コンポーネントデータ間のメッセージ交換によって行い、(B-3) 個々のコンポーネントデータに配送されたメッセージは、上記メッセージ配送プログラムによって処理するようにしたものであり、かかる方法によって、部品を組み合わせしていく過程で、次々に実行し、評価を行い得るものとなる。

【0014】上記アプリケーションプログラムの構成作成支援方法において、好ましくは、上記コンポーネントデータは、プログラム部品のリストを格納する候補リストを有し、この候補リストには、上記プログラム部品エントリとの交換を意図したプログラム部品管理テーブルのプログラム部品を格納し、上記候補リスト内に格納されたプログラム部品のうち、指示された上記候補リスト内の上記プログラム部品を上記プログラム部品エントリに設定するようにしたものであり、候補リストからプログラム部品を容易に選択し得るものとなる。

【0015】上記アプリケーションプログラムの構成作成支援方法において、好ましくは、上記ステップ(A)におけるプログラム部品の指示入力において、識別子としてジョーカー部品が指示された際には、上記ステップ(B-3)において、上記メッセージ配送プログラムは、配送されたメッセージの形式に応じ、上記コンポーネントデータ内の候補リストに格納された個々の部品レコードについて、この部品レコードから取得したメッセージ形式と、配送されたメッセージとを比較し、一致する部品レコードから取得した実行オブジェクトにメッセージを配送するようにしたものであり、かかる方法により、適すると思えるプログラム部品を絞り込み得るものとなる。

【0016】上記アプリケーションプログラムの構成作成支援方法において、好ましくは、上記アプリケーションプログラムの実行が指示された際に検索される上記コンポーネントデータの候補リスト内の部品レコードの検索順序を変更可能とするようにしたものであり、かかる方法により、候補の優先順位を変更し得るものとなる。

【0017】上記アプリケーションプログラムの構成作成支援方法において、好ましくは、上記アプリケーションプログラムの1つの部品組み合わせに対する名称として、実現名称を与え、上記プログラム構成管理テーブルの構成レコードに、上記実現名称を格納する実現名エントリを設け、上記コンポーネントデータ内に、上記部品レコードと上記実現名称の組み合わせを要素とする履歴リストを設け、アプリケーションプログラムの構成の保存が、上記実現名称を付与して指示された場合には、上記プログラム構成管理テーブルの構成レコードの上記実現名称エントリに、この実現名称を記録し、上記コンポ

10

20

30

40

50

ーネントデータ内の上記履歴管理リストに、上記プログラム部品エントリに格納された部品レコードと実現名称を記録するようにしたものであり、変更履歴を管理し得るものとなる。

【0018】上記アプリケーションプログラムの構成作成支援方法において、好ましくは、上記実現名称を表示する複数のノードを、実現の依存関係によってアーク結線した形式によってディスプレイ装置上に表示するようにしたものであり、かかる方法により、アプリケーションプログラムの変更過程を容易に認識し得るものとなる。

【0019】上記アプリケーションプログラムの構成作成支援方法において、好ましくは、上記コンポーネントデータ内に、特定の呼び出しインタフェース情報とその実行オブジェクトを格納するプログラム例エントリを設け、上記メッセージ配送プログラムは、外部からのメッセージ呼び出し形式を参照し、上記プログラム例エントリのインタフェース情報と一致するなら、上記実行オブジェクトにメッセージ配送するようにしたものであり、かかる方法により、未完成部分を含めてアプリケーション全体の動作確認をし得るものとなる。

【0020】上記アプリケーションプログラムの構成作成支援方法において、好ましくは、上記プログラム部品管理テーブルの部品レコードに、接続実績リストを格納するエントリを設け、1つのアプリケーションプログラムが保存された場合、上記プログラム構成管理内テーブルで同じアプリケーションプログラムに含まれたものを、各部品レコードの上記接続実績リストに格納するようにしたものであり、かかる方法により、以前の評価の高い構成を参考とし得るものとなる。

【0021】

【発明の実施の形態】本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法について、図1乃至図30を用いて説明する。

【0022】[1. システム構成] 最初に、全体のシステム構成について、図1を用いて説明する。図1は、本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法において使用するハードウェアと、主なソフトウェア要素を示したシステム構成図である。

【0023】ハードウェアとしては、ディスプレイ装置101、主記憶装置102、CPU装置103、補助記憶装置104、入力装置であるキーボード装置105やマウス装置106が、システムバス107によって接続されている。主記憶装置102に配置されたプログラムが、CPU装置103によって処理される。

【0024】主記憶装置102内には、アプリケーションプログラム構成作成支援プログラム（以下、「構成設定プログラム」とする）123と、構成設定プログラム123が参照するプログラム構成管理テーブル群122とプログラム部品管理テーブル群120が配置されてい

る。

【0025】また、補助記憶装置104内には、過去アプリケーションプログラムとして構築されたもののなかから抽出されるプログラム部品や「Access」のような市販のプログラム部品が格納されている。ここでは、補助記憶装置104内に、プログラム部品A140、プログラム部品B141、プログラム部品C142、プログラム部品D143、プログラム部品E144が格納されているものとする。

10 【0026】構成設定プログラム123は、104内に格納されたプログラム部品A140、…、プログラム部品E144から任意のプログラム部品を組み合わせ、プログラム構成管理テーブル群122およびプログラム部品管理テーブル群120を参照して、アプリケーションプログラム124を作成し、主記憶装置102内に格納する。

【0027】ここで、本実施の形態においては、作成対象となるアプリケーションプログラム124は、プログラム部品140、…、144だけの組み合わせではなく、アプリケーションプログラム124内には、プログラム部品140、…、144そのままではなく、プログラム部品140、…、144を内蔵（またはポインタ処理によって取得可能と）したコンポーネントデータ130、131、132を組み込むようにしている。コンポーネントデータ130、131、132は、内蔵されているアプリケーションプログラムA140、C142、D143の機能を代行する。コンポーネントデータ130、131、132の配置される場所は、プログラム部品が置かれる位置と同じである。従来の手法では、図内に示したように“A”、“C”、“D”などといったプログラム部品のみからアプリケーションプログラムを構成するようにしていた。

【0028】本実施の形態における特徴は、プログラム部品をそのままアプリケーションプログラムで使用するのではなく、コンポーネントデータと呼ぶオブジェクト（データ構造とこれに対するプログラムを内蔵した単位）にプログラム部品を内蔵させ、コンポーネントデータを組み合わせてアプリケーションプログラムを構成することにある。これによって、プログラム部品の変更をコンポーネントデータ内の設定の変更に置き換えることができ、他への影響を最小限に留めることが可能となる。

【0029】コンポーネントデータは、指定されたプログラムの機能単位ごとに割り当てる。ここで、機能単位はプログラム部品の割り当てであるため、コンポーネントデータは、プログラム部品の機能を代行する単位となる。

【0030】ここで、アプリケーションプログラムの内蔵されたコンポーネントデータは、エージェントプログラムの一種であるため、コンポーネントデータ130を

アプリケーションプログラム A140 を内蔵していることから A's-AGT と表し、アプリケーションプログラム C142 を内蔵しているコンポーネントデータ 131 を C's-AGT と表し、アプリケーションプログラム D143 を内蔵しているコンポーネントデータ 132 を D's-AGT と表している。なお、コンポーネントデータが、エージェントプログラム的一种である理由については、後述する。

【0031】プログラム構成管理テーブル群 122 は、機能分割と、それぞれの機能において使用するプログラム部品を記録するものである。また、プログラム部品組合せのうち、過去に実施されたものの履歴情報や履歴間の関係も管理する。プログラム構成管理テーブル群 122 は、3つのテーブルから構成されており、プログラム構成管理テーブル本体 150 と、実現名称を管理する実現管理テーブル 151 と、コンポーネントデータを管理するコンポーネントデータ管理テーブル 152 である。

【0032】また、プログラム部品管理テーブル群 120 は、前述したリポジトリに相当するものである。ここでは、個々のプログラム部品の機能等を記述したプロフィール、他のプログラムからの呼び出し形式（関数名、引数、戻り値等）やプログラム部品の使用実績を管理する。プログラム部品管理テーブル群 120 は、2つのテーブルから構成されており、プログラム部品管理テーブル本体 160 と、メッセージ管理テーブル 161 である。ここで、それぞれのテーブル群 122、120 を分割したのは、管理を容易にすることが目的であり、本実施の形態以外にも分割方法は存在する。

【0033】ディスプレイ装置 101 には、構成設定プログラム 123 の構成設定画面 110 及びアプリケーションプログラム画面 111 が表示される。ユーザは、マウス装置 106 やキーボード装置 105 を用いて、構成設定画面 110 やアプリケーションプログラム画面 111 を操作し、この操作が各プログラムに伝達されて適当な処理が起動される。

【0034】〔2. 構成設定プログラムの概要〕次に、図 2 を用いて、本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いる構成設定プログラム 123 の概要について説明する。図 2 は、本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いる構成設定プログラムの概要を示すフローチャートである。

【0035】構成設定プログラム 123 が起動されると、ステップ 201 において、初期設定処理を行う。初期設定処理の詳細については、図 12 を用いて後述する。

【0036】初期設定が終了すると、ステップ 202 において、ユーザからの入力を受け付けて、入力に応じて処理を起動するイベントフロー型となっている。

【0037】ステップ 203 において、入力がアプリケ

ーションプログラムの機能構成指定であると判断されると、ステップ 204 において、機能構成の設定処理が起動される。機能構成の設定処理の詳細については、図 27 を用いて後述する。

【0038】ステップ 205 において、入力がプログラム部品の構成指定であると判断されると、ステップ 206 において、指定されたプログラム部品に基づくコンポーネントデータ設定処理が起動される。コンポーネントデータ設定処理の詳細については、図 20 を用いて後述する。

【0039】ステップ 207 において、入力がアプリケーションプログラムの実行制御指示であると判断されると、ステップ 208 において、アプリケーションプログラムの実行制御処理が起動される。アプリケーションプログラムの実行制御処理の詳細については、図 16 を用いて後述する。

【0040】ステップ 209 において、入力が終了指示でない場合には、ステップ 202 に戻って入力を受け付け、終了指示である場合には、終了する。

【0041】ここで、構成設定プログラム 123（図 1）は、アプリケーションプログラム 124 内に組み込まれたコンポーネントデータ 130、131、132 と通信対話することで、処理を進める。これは、社会の類似性から、開発対象内にスパイを置き、これに逐一状況を知らせてもらうことに相当する。この意味で、コンポーネントデータ 130、131、132 は、一種のエージェントプログラムである。コンポーネントデータ 130、131、132 のエージェントとしての役割は、内部に有するメッセージ配送プログラムが果たしている。図 9 に示すメッセージ配送プログラム 2000 の詳細については、図 17、18 を用いて後述する。

【0042】〔3. 本実施の形態で使用する開発対象アプリケーションプログラム〕次に、実際にアプリケーションを作成し、変更していく場面を想定し、その中で構成設定プログラム（図 2；詳細は、図 16、図 20、図 27）とメッセージ配送プログラム（図 9：処理手順は、図 17、図 18）がどのように通信し、設計者を支援していくかを説明することによって、本実施の形態によるアプリケーションプログラムの構成作成支援方法の効果について、図 3、図 4、図 5 を用いて説明する。

【0043】ここで述べる例としては、過去に作成されたアプリケーションプログラム（図 3）の各プログラム部品の変更を行い（図 4）、新しいプログラム（図 5）を作成する過程を述べる。変更の道筋をはじめに述べ、この中で構成設定処理がどのように動作するかを説明する。

【0044】図 3 は、本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法を説明するための過去に作成されたアプリケーションプログラムの構成図であり、図 4 は、本発明の一実施の形態によるア

アプリケーションプログラムの構成作成支援方法による変更の過程を示す説明図であり、図5は、本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法により最終的に作成されたアプリケーションプログラムの構成図である。

【0045】図3（A）は、全体のプログラム構成を示し、図3（B）は、クライアント側のプログラムの一部の詳細なプログラム構成を示している。

【0046】図3に示すアプリケーションプログラムの目的は、例えば、在庫管理、受発注処理や販売データの解析を行うことである。

【0047】システムは、図3（A）に示すように、データベース処理を行う販売管理サーバ計算機に格納された販売管理サーバプログラム300と3台の販売管理クライアント計算機に格納された販売管理クライアントプログラム301、302、303から構成され、かかるシステムで動作した実績がある。販売管理サーバプログラム300と販売管理クライアントプログラム301、302、303とが通信して処理を進めていく。

【0048】販売管理サーバプログラム300内の主な処理は、サーバーメイン（Server Main）プログラム310が担当しており、処理単位で分割されたプログラム部品の動作を制御する。販売管理サーバプログラム300側のプログラム部品としては、在庫管理データベースである在庫マスタデータベース（Stock DB）311、受発注データのデータベースである受発注マスタデータベース（Order DB）312、販売データ解析データのデータベースである解析マスタデータベース（Analysis DB）313、それぞれのデータベースのクライアント呼び出しインタフェースを担当する在庫管理（Stock Mgr（Manager））314、受発注管理（Order Mgr）315、販売データ解析処理（Analysis Mgr）316の各プログラム部品から構成されている。

【0049】販売管理クライアントプログラム301の主な処理は、クライアントメイン（Client Main）プログラム323が担当しており、処理単位で分割されたプログラム部品の動作を制御する。販売管理クライアントプログラム301側のプログラム部品としては、在庫画面（Stock Clt（Client））322、受発注画面（Order Clt）323、販売データ解析画面（Analysis Clt）324のプログラム部品から構成されている。

【0050】さらに、図3（B）に示すように、販売データ解析画面324は、販売データ表示330、販売予測表示331、予測パラメータ設定332、販売予測333から構成されている。販売データ表示330は、さらに、購買地域別表示340、購買年代別表示341、購買季節別表示342から構成されている。

【0051】図3に示す構成を出発点として、新しい目

的に一致するアプリケーションプログラムを作成する。新しいシステム構成として、クライアントが10台になり、また、クライアント内の画面デザイン（画面レイアウト、操作方法）が変更される。ここで、クライアント台数の変更は、データベースのアクセス性能の問題を引き起こす。ここで重要なことは、サーバ側、クライアント側ともにプログラム部品の内容、構成の変更が必要となることである。

【0052】次に、図4を用いて、変更の過程の一例について説明する。最初に図3に示した初期構成に対して、管理部品の変更を行うことにより、代案1を作成し、データベース（DB）の変更を行うことにより、代案2を作成、画面レイアウトの変更を行うことにより、代案3を作成し、画面操作方法の変更を行うことにより、代案4を作成していくものとする。代案1、代案2、代案3、代案4からなる第1の系統の変更を行いながら、検討を進めたところ、機能を満足しないことが判明した。

【0053】そこで、初期構成に戻り、サブデータベース（DB）を設置する代案5を作成し、管理部品の変更を行うことにより、代案6を作成し、第1の系統の最終案である代案4（DB構造の変更、画面レイアウトの変更、画面操作方法の変更を含む）を参照して、部品の横流しをして、最終案として代案7を作成する。第1の系統では、ローカルなデータベースを配置しなかったことが原因で設計変更が発生し、第2の系統で構成を変更して再検討したことを表している。

【0054】図5は、図4の検討結果により作成された最終案である代案7の構成を示している。図5（A）は、全体のシステム構成及びプログラム構成を示し、図5（B）は、クライアント側のプログラムの一部の詳細なプログラム構成を示している。

【0055】システムは、図5（A）に示すように、データベース処理を行う販売管理サーバ計算機に格納された販売管理サーバプログラム500と10台の販売管理クライアント計算機に格納された販売管理クライアントプログラム501、502、…、509、510から構成される。販売管理サーバプログラム500と販売管理クライアントプログラム501、502、…、509、510とが通信して処理を進めていく。

【0056】販売管理サーバプログラム500内のプログラム部品は、サーバーメイン（Server Main）プログラム510と、在庫管理データベースである在庫マスタ511、受発注データのデータベースである受発注マスタ512、販売データ解析データのデータベースである解析マスタ513、それぞれのデータベースのクライアント呼び出しインタフェースを担当する在庫管理514、受発注管理515、販売データ解析処理516から構成されている。これらは、基本的には、図3のServer Mainプログラム310、在庫マ

タ311, 受発注マスタ312, 解析マスタ313, 在庫管理314, 受発注管理315, 販売データ解析処理316と対応するものであるが、それぞれ、プログラム部品の変更(交換)がなされている。

【0057】また、販売管理クライアントプログラム501のプログラム部品としては、サブデータベースとして、在庫サブデータベース(Stock Sub)525, 受発注サブデータベース(Order Sub)526, 解析サブデータベース(Analysis Sub)527が追加され、クライアントメイン(Client Main)プログラム523と、在庫画面522, 受発注画面523, 販売データ解析画面524のプログラム部品が変更されている。

【0058】さらに、図5(B)に示すように、販売データ解析画面524は、販売データの表示及び予測を行う販売データ・予測表示530, 予測パラメータ設定532, 販売予測533に変更されている。販売データ・予測表示530は、さらに、変更された購買地域別表示540, 購買年代別表示541, 購買季節別表示542から構成されている。

【0059】以下の説明では、図3に示した初期構成から図5に示した代案7に至るまでの変更を、構成設定プログラム123(図1)が、いかに支援するかについて説明する。

【0060】[4. 初期状態における管理テーブル群]次に、図1に示したプログラム構成管理テーブル群122の中のプログラム構成管理テーブル本体150, 実現管理テーブル151, コンポーネントデータ管理テーブル152の内容の初期状態, コンポーネントデータ130の構成及びプログラム部品管理テーブル群120の中のプログラム部品管理テーブル本体160, メッセージ管理テーブル161の内容の最終状態について、図6, 図7, 図8, 図9, 図10, 図11を用いて説明する。図6は、本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いるプログラム構成管理テーブル150の構成図であり、図7は、本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いる実現管理テーブル151の構成図であり、図8は、本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いるコンポーネントデータ管理テーブル152の構成図であり、図9は、本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いるコンポーネントデータ130の構成図であり、図10は、本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いるプログラム部品管理テーブル160の構成図であり、図11は、本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いるメッセージ管理テーブル161の構成図である。

【0061】なお、図6, 図7, 図8, 図10, 図11

は、いずれも、図3に示したアプリケーションプログラムが過去に作成されている場合の内容を示している。

【0062】図6において、プログラム構成管理テーブル150(図1)は、識別子エントリ601, プログラム名エントリ602, 実現名リストエントリ603の各エントリから構成されるレコードを管理する。

【0063】ここで、開発対象となるシステム(図3)は、販売管理サーバプログラムと販売管理クライアントプログラムの2つのプログラムから構成されているので、対応する2つのレコードが登録されている。プログラム名エントリ602の「販売管理サーバ」のプログラムは、識別子エントリ601のフィールドに「1」として登録されており、この「販売管理サーバ」に対応する実現名リストエントリ603のフィールドには、1つだけの実現名「A1」が登録されている。プログラム名エントリ602の「販売管理クライアント」のプログラムは、識別子エントリ601のフィールドに「2」として登録されており、この「販売管理クライアント」に対応する実現名リストエントリ603のフィールドにも、1つだけの実現名「A100」が登録されている。

【0064】ここで、実現名は、別に管理されるテーブルである実現管理テーブル151の識別子となっている。実現管理テーブル151の構成については、図7を用いて説明する。

【0065】図7において、実現管理テーブル151(図1)は、部品の組み合わせを管理するものであり、識別子エントリ701, コンポーネントデータリストエントリ702, 参照エントリ703, コンポーネント外リストエントリ704の各エントリから構成されるレコードを管理する。

【0066】識別子エントリ701のフィールドに記録される識別子は、図6の実現名リストエントリ603の実現名を識別子とするものであり、例えば、図6に示した実現名「A1」が識別子となっている。

【0067】コンポーネントデータリストエントリ702は、コンポーネントデータ管理テーブル152(図8)で管理されるコンポーネントデータの識別子のリストである。例えば、識別子エントリ701のフィールドに識別子「A1」で識別される実現名においては、コンポーネントリストエントリ702のフィールドには、「1, 2, 3, 4, 5, 6」が登録されている。これらの「1, ..., 6」は、後述するコンポーネントデータ管理テーブル152(図8)で管理されるコンポーネントデータの識別子である。「1, 2, 3, 4, 5, 6」で識別されるコンポーネントデータは、それぞれ、「在庫管理」、「発注管理」、「解析管理」、「在庫DB」、「発注DB」、「解析DB」を機能単位としているものであり、「販売管理サーバ」の1つの実現組み合わせを示している。

【0068】参照エントリ703は、実現組み合わせが

識別子で表されるどの実現を変更したものを示すものである。図7に示す初期構成の状態では、参照エントリ703のフィールドの記録は、「-」となっており、これは、どの実現も参照していないことを表している。プログラムの変更により、新しいプログラムが作成された場合には、その新しいプログラムの変更の元となったプログラムの実現が登録される。

【0069】コンポーネント外リストエントリ704は、実現を構成するプログラム部品のうち、部品化されていないものを表している。プログラム部品の定義は、使用するプラットフォームによって異なるが、一般に部品全体の初期割り付け、イベント処理を行うプログラムが部品外として扱われる。本実施の形態でも、この意味でプログラム部品とそうでないプログラムを区別している。識別子「A1」に対してコンポーネント外リストエントリ704のフィールドに登録されたコンポーネント外リスト「Main 0」は、図3のサーバメイン（Server Main）プログラム310のことである。

【0070】識別子エントリ701のフィールドに登録された識別子「A100」は、図6に示した実現名「A100」が識別子となっている。識別子エントリ701のフィールドに識別子「A100」で識別される実現名においては、コンポーネントリストエントリ702には、「7、8、9」が登録されている。これらの「7、8、9」は、後述するコンポーネントデータ管理テーブル152（図8）で管理されるコンポーネントデータの識別子である。「7、8、9」で識別されるコンポーネントデータは、それぞれ、「在庫画面」、「受発注画面」、「販売データ解析画面」を機能単位としているものであり、「販売管理クライアント」の1つの実現組み合わせを示している。

【0071】参照エントリ703のフィールドの記録は、図7に示す初期構成の状態では、「-」となっており、これは、どの実現も参照していないことを表している。

【0072】識別子「A100」に対してコンポーネント外リストエントリ704に登録されたコンポーネント外リスト「Main 100」は、図3のクライアントメイン（Client Main）プログラム321のことである。

【0073】図8において、コンポーネントデータ管理テーブル152は、識別子エントリ801、コンポーネント名エントリ802及び図示しない機能（記述）エントリの各エントリから構成されるレコードを管理する。

【0074】識別子エントリ801のフィールドに登録された識別子は、図7のコンポーネントデータリストエントリ702の実現名を識別子とするものであり、例えば、図7に示した実現名「1、2、3、4、5、6」が識別子となっている。

【0075】コンポーネント名エントリ802の実現名は、コンポーネントデータを識別する名称となっている。

【0076】識別子エントリ801のフィールドに登録された識別子「1」のコンポーネント名は、「在庫管理エージェント（Stock Mgr Agent）」であり、図3に示したプログラム部品である在庫管理（Stock Mgr）314に対するコンポーネントデータを識別するものとなっている。コンポーネントデータの詳細な構成については、図9を用いて後述する。

【0077】識別子エントリ801のフィールドに登録された識別子「2」のコンポーネント名は、「受発注管理エージェント（Order Mgr Agent）」であり、図3に示したプログラム部品である受発注管理（Order Mgr）315に対するコンポーネントデータを識別するものである。

【0078】識別子エントリ801のフィールドに登録された識別子「3」のコンポーネント名は、「販売データ解析処理エージェント（Analysis Mgr Agent）」であり、図3に示したプログラム部品である販売データ解析処理（Analysis Mgr）316に対するコンポーネントデータを識別するものである。

【0079】識別子エントリ801のフィールドに登録された識別子「4」のコンポーネント名は、「在庫マスタデータベースエージェント（Stock DB Agent）」であり、図3に示したプログラム部品である在庫マスタデータベース（Stock DB）311に対するコンポーネントデータを識別するものである。

【0080】識別子エントリ801のフィールドに登録された識別子「5」のコンポーネント名は、「受発注マスタデータベースエージェント（Order DB Agent）」であり、図3に示したプログラム部品である受発注マスタデータベース（Order DB）312に対するコンポーネントデータを識別するものである。

【0081】識別子エントリ801のフィールドに登録された識別子「6」のコンポーネント名は、「解析マスタデータベースエージェント（Analysis DB Agent）」であり、図3に示したプログラム部品である解析マスタデータベース（Analysis DB）313に対するコンポーネントデータを識別するものである。

【0082】識別子エントリ801のフィールドに登録された識別子「7」のコンポーネント名は、「在庫画面エージェント（Stock Clt Agent）」であり、図3に示したプログラム部品である在庫画面（Stock Clt）322に対するコンポーネントデータを識別するものである。

【0083】識別子エントリ801のフィールドに登録

された識別子「8」のコンポーネント名は、「受発注画面エージェント (Order Clt Agent)」であり、図3に示したプログラム部品である受発注画面 (Order Clt) 323に対するコンポーネントデータを識別するものである。

【0084】識別子エン트리801のフィールドに記録された識別子「9」のコンポーネント名は、「販売データ解析画面エージェント (Analysis Clt Agent)」であり、図3に示したプログラム部品である販売データ解析画面 (Analysis Clt) 324に対するコンポーネントデータを識別するものである。

【0085】次に、図9を用いて、コンポーネントデータ130の概略の構成について説明する。なお、詳細な機能については、各処理の説明に併せて後述する。

【0086】ここで、コンポーネントデータ130は、永続的なデータであり、構成設定プログラム123で処理を受けた状態のまま、補助記憶装置104に格納される。

【0087】コンポーネントデータ130は、メッセージ配送プログラム900、選択プログラム部品エン트리910、候補リスト911、プログラム例エン트리912、選択プログラム履歴リスト913、ログテーブル914から構成される。

【0088】プログラム部品間のメッセージ通信は、コンポーネントデータ130間の通信となる。この通信は、メッセージ配送プログラム900が処理する。メッセージ配送プログラム900は、選択プログラム部品エン트리910、候補リスト911の要素、プログラム例エン트리912のどれかから取得されるプログラムにメッセージを配送する。これを決定するのはプログラマであり、決定するための画面インタフェースを用意している。

【0089】選択プログラム部品エン트리910は、選択された1個のプログラム部品を格納するエン트리であり、ここにプログラム部品を直接記憶するか、プログラム部品へのポインタを格納する。Functions 920は、機能リストを格納するエン트리であり、Slots 921は、ローカルなデータのリストを格納するエン트리である。

【0090】候補リスト911は、選択されるプログラム部品の候補となる複数のプログラム部品を格納するエン트리であり、複数のエン트리930、931、932から構成されている。

【0091】プログラム例エン트리912は、メッセージに応じたユーザの設定するプログラム例を格納するエン트리である。

【0092】選択プログラム履歴リスト913は、選択プログラム部品エン트리910として選択されたプログラム部品の中で、評価された結果、記録に値するものを

格納するエン트리である。選択プログラム履歴リスト913は、プログラム部品識別子950と実現名951から構成され、プログラム部品識別子950には、記録に値すべき選択されたプログラム部品の識別子が登録され、実現名951に、登録された識別子の実現名が登録される。

【0093】ログテーブル914は、プログラム部品の変更を行った後、プログラムを実行した際のメッセージ呼び出しの成功、不成功の結果が記録される。ログテーブル914は、メッセージ960、送信元961及び結果962から構成される。メッセージ960には、送信元から送られるメッセージの内容が記録され、送信元961には、そのメッセージを送った送信元が記録され、結果962には、メッセージ送信の成功、不成功の結果が記録される。

【0094】次に、図10を用いて、プログラム部品管理テーブル本体160の状態について説明する。但し、図10に示すの状態は初期状態ではなく、最終案まで開発が完了した時のものである。図の上では、初期状態から識別子2、3、14が追加されて最終状態になっている。

【0095】プログラム部品管理テーブル160は、個々のプログラム部品の機能等を記述したプロフィールやプログラム部品の使用実績を管理するものである。プログラム部品管理テーブル本体160は、識別子エン트리1001、プログラム部品を機能分類したプログラム部品分類エン트리1002、プログラム名エン트리1003、接続実績リストエン트리1004の各エン트리から構成されるレコードを管理する。

【0096】図3に示した在庫管理 (Stock Mgr) 314のプログラム部品は、プログラム名エン트리1003のフィールドに実現名「Stock Mgr」として登録されており、その識別子は、識別子エン트리1001のフィールドに識別子「1」として登録されている。

【0097】この「Stock Mgr」は、在庫管理に関するプログラム部品であるため、プログラム部品分類エン트리1002のフィールドには、「在庫管理」とそのプロフィールが登録されている。プログラム名エン트리1003の実現名「Stock Mgr_M」や「Stock Mgr_MM」のプログラム部品は、実現名「Stock Mgr」のプログラム部品を変更したものであるため、プログラム部品分類エン트리1002のフィールドには、「在庫管理」と登録されている。

【0098】接続実績リストエン트리1004は、そのプログラムがどのプログラム部品と接続されて使用された実績があるかを示したものである。例えば、識別子「1」の「Stock Mgr」のプログラム部品は、識別子95、96、520、522と接続された実績を有することを示している。

【0099】識別子エン트리1001の識別子「13」は、図3に示した受発注管理 (Order Mgr) 315のプログラム部品の識別子であり、プログラム部品分類エン트리1002には、「受発注管理」とそのプロフィールが登録されている。

【0100】プログラム名エン트리1003に実現名「Order Mgr」として登録されており、接続実績リストエン트리1004には、識別子124, 125, 527と接続された実績を有することを示している。

【0101】識別子エン트리1001の識別子「14」は、プログラム名エン트리1003に登録されているように、「Order Mgr」プログラムを変更した「Order Mgr」プログラムであり、プログラム部品分類エン트리1002には、「受発注管理」とそのプロフィールが登録されている。接続実績リストエン트리1004には、識別子125, 530と接続された実績を有することを示している。

【0102】識別子エン트리1001の識別子「26」は、図3に示した販売データ解析処理 (Analysis Mgr) 316のプログラム部品の識別子であり、プログラム部品分類エン트리1002には、「販売データ解析処理」とそのプロフィールが登録されている。プログラム名エン트리1003に実現名「Analysis Mgr」として登録されており、接続実績リストエン트리1004には、識別子127, 535と接続された実績を有することを示している。

【0103】識別子エン트리1001の識別子「95」は、図3に示した在庫管理マスタデータベース (Stock DB) 311のプログラム部品の識別子であり、プログラム部品分類エン트리1002には、「在庫管理マスタ」とそのプロフィールが登録されている。プログラム名エン트리1003に実現名「Stock DB」として登録されており、接続実績リストエン트리1004には、識別子1, 2, 3と接続された実績を有することを示している。

【0104】識別子エン트리1001の識別子「124」は、図3に示した受発注管理マスタデータベース (Order DB) 312のプログラム部品の識別子であり、プログラム部品分類エン트리1002には、「受発注管理マスタ」とそのプロフィールが登録されている。プログラム名エン트리1003に実現名「Order DB」として登録されており、接続実績リストエン트리1004には、識別子13と接続された実績を有することを示している。

【0105】識別子エン트리1001の識別子「127」は、図3に示した販売データ解析マスタデータベース (Analysis DB) 313のプログラム部品の識別子であり、プログラム部品分類エン트리1002には、「販売データ解析マスタ」とそのプロフィールが

登録されている。プログラム名エン트리1003に実現名「Order DB」として登録されており、接続実績リストエン트리1004には、識別子28と接続された実績を有することを示している。

【0106】識別子エン트리1001の識別子「520」は、図3に示した在庫管理画面 (Stock Clt) 322のプログラム部品の識別子であり、プログラム部品分類エン트리1002には、「在庫管理画面」とそのプロフィールが登録されている。プログラム名エン트리1003に実現名「Stock Clt」として登録されており、接続実績リストエン트리1004には、識別子1と接続された実績を有することを示している。

【0107】識別子エン트리1001の識別子「527」は、図3に示した受発注管理画面 (Order Clt) 323のプログラム部品の識別子であり、プログラム部品分類エン트리1002には、「受発注管理画面」とそのプロフィールが登録されている。プログラム名エン트리1003に実現名「Order Clt」として登録されており、接続実績リストエン트리1004には、識別子13, 15と接続された実績を有することを示している。

【0108】識別子エン트리1001の識別子「535」は、図3に示した販売データ解析画面 (Analysis Clt) 324のプログラム部品の識別子であり、プログラム部品分類エン트리1002には、「販売データ解析画面」とそのプロフィールが登録されている。プログラム名エン트리1003に実現名「Analysis Clt」として登録されており、接続実績リストエン트리1004には、識別子28と接続された実績を有することを示している。

【0109】識別子エン트리1001の識別子「Main 0」は、図3に示したサーバメイン (Server Main) 310のプログラム部品の識別子であり、プログラム部品分類エン트리1002には、「販売管理サーバ制御」とそのプロフィールが登録されている。プログラム名エン트리1003に実現名「Server Main」として登録されており、接続実績リストエン트리1004には、識別子1, 13, 28, 95, 124, 127と接続された実績を有することを示している。

【0110】識別子エン트리1001の識別子「Main 100」は、図3に示したクライアントメイン (Client Main) 321のプログラム部品の識別子であり、プログラム部品分類エン트리1002には、「販売管理クライアント制御」とそのプロフィールが登録されている。プログラム名エン트리1003に実現名「Client Main」として登録されており、接続実績リストエン트리1004には、識別子520, 527, 535と接続された実績を有することを示している。

【0111】接続実績リストエントリ1004に記録された接続実績の情報は、新しいアプリケーションの構成の参考とすることができる。即ち、プログラム部品レコードに利用実績を記録することで、以前に使用されたアプリケーションと、1つの部品がどの部品と組み合わせて使用されたかが記録される。これによって、相性の良い、実績のある組み合わせ、換言すると、評価が高い構成を再現することことで、これを参考として、新しいアプリケーションの構成の参考とすることができる。

【0112】各プログラム部品の実際の使用方法として10は、各プログラム部品の識別子を外部からメッセージ（関数・メソッドコール）によって呼び出すことによって行う。

【0113】次に、図11を用いて、メッセージ管理テーブル161の構成について説明する。

【0114】メッセージ管理テーブル161は、プログラム部品毎に、どのようなメッセージを処理可能かを管理するテーブルである。メッセージ管理テーブル161は、識別子エントリ1101、プログラム部品テーブルの識別子エントリ1102、メッセージ名称エントリ1103、引数の数（Arg（Argument） Count）エントリ1104、第1引数から第3引数までのデータの型の第1引数（Argument 1）エントリ1105、第2引数（Argument 2）エントリ1106、第3引数（Argument 3）エントリ1107の各エントリから構成されるレコードを管理する。

【0115】例えば、識別子1のレコードで管理するメッセージは、プログラム部品管理テーブル本体160の識別子エントリ1001のフィールドが「1」のプログラム部品1（Stock Mgr）のもので、メッセージ名称エントリ1103のフィールドに登録されているように「Add」のメッセージインターフェースを有しており、引数の数（Arg Count）エントリ1104のフィールドに登録されているように、引数「2」をとり、第1引数（Argument 1）エントリ1105のフィールドに登録されているように、第1引数はItem型であり、第2引数（Argument 2）エントリ1106のフィールドに登録されているように、第2引数の型は整数型であり、従って、Add（Item, int）の形式で呼び出されることを示している。

【0116】また、プログラム部品管理識別子1302のフィールドに登録された識別子「1」の「Stock Mgr」は、「Add」、「Remove」、「Refer」などのメッセージインターフェースを有している。

【0117】本実施の形態によれば、プログラム部品レコードに利用実績を接続実績リストに登録することで、以前に使用されたアプリケーションと、1つの部品がど

の部品と組み合わせて使用されたかが判るため、相性の良い、実績のある組み合わせ、換言すると、評価が高い構成を再現することことで、これを参考として、新しいアプリケーションの構成の参考とすることができる。

【0118】[5. コンポーネントデータを含むアプリケーションプログラム初期構成の作成と実行] 次に、コンポーネントデータを含むアプリケーションプログラム初期構成の作成と実行について、図2、図6、図7、図9、図10、図12、図13、図14、図15を用いて説明する。

【0119】図12は、本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いる初期設定処理ステップ201の詳細フローチャートであり、図13は、本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法において表示する構成設定画面110の詳細な画面構成の構成図であり、図14は、本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法において構成された初期のアプリケーションプログラムである販売管理サーバプログラムの構成図であり、図15は、本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法において構成された初期のアプリケーションプログラムである販売管理クライアントプログラムの構成図である。

【0120】構成設定プログラム123が起動されると、初めに、図2のステップ201の初期設定処理が起動される。初期設定処理では、図3に示したアプリケーションプログラムの初期構成を指定し、これを変更していくことを入力とする。ステップ201の詳細については、図12に示したあるため、以下の説明は、図12に示したフローチャートに基づいて行う。

【0121】ステップ1200において、初めに、図1に示したディスプレイ装置101に構成設定画面110を表示する。構成設定画面110の詳細な画面構成について、図13を用いて説明する。なお、図13は、設計がある程度進んだ状態を表しており、初期画面との相違については以下に説明する。

【0122】構成設定画面110は、いくつかのサブ画面に分割され、サブ画面に設計情報を表示し、その内容を変更することで設計を進める。サブ画面としては、アプリケーションプログラム名を表示するプログラム（Program）サブ画面1301、アプリケーションプログラムを構成する複数のコンポーネントデータを表示するコンポーネント（Components）サブ画面1302、各コンポーネントデータに対する複数のプログラム部品候補を表示する候補（Candidates）サブ画面1303、メッセージの送信記録を表示するメッセージログ（Message Log）サブ画面1325、メッセージに応じたユーザの設定するプログラム例を表示するプログラム例（Example）サブ

画面1332から構成されている。

【0123】なお、コンポーネント (Component) サブ画面1302の中は、プログラム部品分類を表示する分類 (Classification) 画面1315とプログラム部品のプログラム名を表示するプログラム名 (ID Name) 画面1316に分割されている。

【0124】さらに、プログラム (Program) サブ画面1301の下には、アプリケーションプログラムの追加を行うための追加 (Add) ボタン1306と、プログラムの削除を行うための削除 (Remove) ボタン1307が表示される。コンポーネント (Components) サブ画面1302の下には、プログラム部品の追加を行うための追加 (Add) ボタン1308と、プログラム部品の削除を行うための削除 (Remove) ボタン1309が表示される。

【0125】候補 (Candidates) サブ画面1303の下には、過去実現例からの参照を行うための履歴 (History) ボタン1310、候補プログラムの追加を行うための追加 (Add) ボタン1311及び候補プログラムの削除を行うための削除 (Remove) ボタン1312が表示される。候補 (Candidates) サブ画面1303の上には、候補リストとして表示された複数のプログラム部品候補のリスト上の実行優先順位を変更するための上矢印ボタン1304及び下矢印ボタン1305が表示される。

【0126】メッセージログ (Message Log) サブ画面1325の下には、アプリケーションプログラムの実行を指示するための実行 (Run) ボタン1320と、実行中止を指示するための実行中止 (Stop) ボタン1321が表示される。

【0127】プログラム例 (Example) サブ画面1332の上は、ユーザが設定したプログラム例の登録を指示するための登録 (Register) ボタン1330と、設定したプログラム例を削除するための削除 (Remove) ボタン1331が表示される。

【0128】さらに、構成設定画面110の最下部には、プログラム部品の評価結果として記録に値するものが構成された場合に、新しい構成の記録の指示を行う記録 (Record) ボタン1340と、構成設定プログラムから抜け出すための出口 (Exit) ボタン1341が表示される。

【0129】なお、ステップ1200の立ち上げ時において、構成設定画面を表示した段階では、プログラム (Program) サブ画面1301、コンポーネント (Components) サブ画面1302、メッセージログ (Message Log) サブ画面1325に図示してある情報は、まだ、表示されていない。

【0130】ステップ1201において、作成対象とするアプリケーションプログラムの元となるアプリケーション

ョンプログラムの実現名称を受け付ける。

【0131】ここでは、図3に示した初期構成から、販売管理サーバプログラムと販売管理クライアントプログラムの2つのプログラムが必要であることから、2つのプログラム (販売管理サーバ、販売管理クライアント) の実現名「A1」、「A100」を入力する。これらの名称は、別に表示されるダイアログウィンドウを用いて入力される。

【0132】ステップ1202において、実現名称からコンポーネントデータリストを取得し、各コンポーネントデータを主記憶装置上に展開し、アプリケーションプログラムを構成する。

【0133】即ち、入力された実現名「A1」、「A100」に応じて、プログラム構成管理テーブル150 (図6) のレコードを取得する。実現名「A1」は、図6の識別子エントリ601の識別子「1」に対応し、また、実現名「A100」は、図6の識別子エントリ601の識別子「2」に対応するため、これらのレコードを取得する。

【0134】さらに、それぞれの実現名称を有する実現管理テーブル151 (図7) のレコードを取得し、このレコードのコンポーネントデータリストを取得する。図7の識別子エントリ701に「A1」の実現名を持つレコードは、コンポーネントデータ「1, 2, 3, 4, 5, 6」である。識別子エントリ701のフィールドに「A100」の実現名を持つレコードは、コンポーネントデータ「7, 8, 9」である。

【0135】次に、これらに対応するコンポーネントデータ管理テーブル152 (図8) のレコードを取得する。図8の識別子エントリ801のフィールドに「1, 2, 3, 4, 5, 6」の実現名を持つレコードは、それぞれ、「在庫管理エージェント (Stock Mgr Agent)」, 「受発注管理エージェント (Order Mgr Agent)」, 「販売データ解析処理エージェント (Analysis Mgr Agent)」, 「在庫マスタデータベースエージェント (Stock DB Agent)」, 「受発注マスタデータベースエージェント (Order DB Agent)」, 「解析マスタデータベースエージェント (Analysis DB Agent)」である。また、識別子エントリ801に「7, 8, 9」の実現名を持つレコードは、それぞれ、「在庫画面エージェント (Stock Clt Agent)」, 「受発注画面エージェント (Order Clt Agent)」, 「販売データ解析画面エージェント (Analysis Clt Agent)」である。

【0136】さらに、これらのコンポーネント名から、それぞれのコンポーネントデータ130を取得し、アプリケーションプログラムを構成する。

【0137】コンポーネントデータ130内には、選択

プログラム履歴リスト913 (図9) が格納されており、選択プログラム履歴リスト913のテーブルには、実現名エントリ951に実現名が記録されている。そこで、入力された実現名称と一致するレコードを取得する。そして、選択プログラム部品エントリ910に、プログラム部品管理テーブル本体160 (図10) のプログラム部品識別子エントリ1001のフィールドから取得したプログラム部品を設定する。例えば、「在庫管理エージェント (Stock Mgr Agent)」のコンポーネントデータ130では、入力された実現名称から「Stock Mgr」のプログラム部品を取得し、これを選択プログラム部品エントリ910のフィールドに設定する。同様に、他のコンポーネントデータについても、プログラム部品を取得し、選択プログラム部品に設定する。

【0138】以上の処理によって、初期のアプリケーションプログラムが構成される。

【0139】構成された初期のアプリケーションプログラムの構造について、図14、図15を用いて説明する。図14は、初期のアプリケーションプログラムである販売管理サーバプログラムの構成図であり、図15は、初期のアプリケーションプログラムである販売管理クライアントプログラムの構成図である。

【0140】図14において、販売管理サーバプログラム300は、プログラム部品を内蔵したコンポーネントデータである在庫管理エージェント (Stock Mgr Agent) 1410、受発注管理エージェント (Order Mgr Agent) 1411、販売データ解析処理エージェント (Analysis Mgr Agent) 1412、在庫マスタデータベースエージェント (Stock DB Agent) 1413、受発注マスタデータベースエージェント (Order DB Agent) 1414、解析マスタデータベースエージェント (Analysis DB Agent) 1415の集合と、その他のプログラムであるサーバーメイン (Server Main) プログラム310から構成される。

【0141】コンポーネントデータ130の内部の構成は、図9に示したとおりであり、初期状態のコンポーネントデータ130においては、例えば、在庫管理エージェント (Stock Mgr Agent) 1410の内部構成は、図14に例示したように、選択プログラム部品エントリのフィールドには、「Stock Mgr」が格納され、候補リストエントリ、プログラム例エントリ、選択プログラム履歴リストは、「空」の状態となっている。

【0142】他のコンポーネントデータ1411、1412、1413、1414、1415についても、同様にして、選択プログラム部品エントリのフィールドには、それぞれの初期のプログラム部品が格納され、候補

リストエントリ、プログラム例エントリ、選択プログラム履歴リストは、「空」の状態となっている。

【0143】図15において、販売管理クライアントプログラム301は、プログラム部品を内蔵したコンポーネントデータである在庫画面エージェント (Stock Clt Agent) 1510、受発注画面エージェント (Order Clt Agent) 1511、販売データ解析画面エージェント (Analysis Clt Agent) 1512の集合と、その他のプログラムであるクライアントメイン (Client Main) プログラム321から構成される。

【0144】例えば、在庫管理エージェント (Stock Mgr Agent) 1410の内部構成は、図15に例示したように、選択プログラム部品エントリのフィールドには、「Stock Clt」が格納され、候補リストエントリ、プログラム例エントリ、選択プログラム履歴リストは、「空」の状態となっている。

【0145】他のコンポーネントデータ1511、1512についても、同様にして、選択プログラム部品エントリのフィールドには、それぞれの初期のプログラム部品が格納され、候補リストエントリ、プログラム例エントリ、選択プログラム履歴リストは、「空」の状態となっている。

【0146】次に、ステップ1203において、ステップ1202において構成されたアプリケーションプログラムの構成を、構成設定画面110 (図13) に表示する。

【0147】このステップ1203によって、図13に示す画面のうち、Programサブ画面1301、Componentsサブ画面602に情報が表示される。

【0148】図13における画面表示の意味するところは、次のとおりである。Programサブ画面1301に表示される「販売管理サーバ」、「販売管理クライアント」は、このシステムが、販売管理サーバプログラムと販売管理クライアントプログラムの2つのプログラムから構成されることを意味している。

【0149】これらの2つのプログラムのうち、「販売管理サーバ」に付された下線は、販売管理サーバプログラム選択されていることを示している。選択されたプログラムのプログラム部品の構成がComponentsサブ画面602に表示されている。

【0150】Componentsサブ画面602には、販売管理サーバプログラムが6つの部品機能群から構成されることを示している。6つの部品機能群のうち、在庫管理の部品機能は、「Stock Mgr」と名前のついたプログラム部品によって実装されている。受発注管理の部品機能は、「Order Mgr」というプログラム部品によって実装されており、販売データ解析の部品機能は、「Analysis Mgr」というプ

ログラム部品によって実装されている。また、在庫マスタの部品機能は、「Stock DB」というプログラム部品によって実装されており、受発注マスタの部品機能は、「Order DB」というプログラム部品によって実装されており、解析マスタの部品機能は、「Analysis DB」というプログラム部品によって実装されていることがわかる。

【0151】「在庫管理」「Stock Mgr」に下線が付されており、このプログラム部品が選択されているので、このプログラム部品に実装の候補となるプログラム部品があるときは、候補 (Candidates) サブ画面1303に候補リストが表示される。しかしながら、ここでは、何も候補にあがっていない。これは始めたばかりで、他に替わる候補部品が指示されていないためである。

【0152】図13の表示内容は、次のようにして変更することができる。なお、この処理については、繁雑になるため構成設定処理の処理フロー内には明示していない。

【0153】プログラム (Program) サブ画面1301内のプログラム名をマウス装置106で選択することで、このプログラムを構成するコンポーネントデータが、コンポーネント (Components) サブ画面1302に表示される。コンポーネント (Components) サブ画面1302内のコンポーネント名または部品名をマウス装置106で選択することで、このコンポーネントの代替候補が、候補 (Candidates) サブ画面1303に表示される。コンポーネント (Components) サブ画面1302内での選択と、候補 (Candidates) サブ画面1303表示されたプログラム部品の選択は、その部品を選択したこととなり、後述するMessage Logテーブル914内の表示や、プログラム例1332の設定の対象を選択する操作となる。

【0154】プログラムの追加、削除は、追加 (Add) ボタン1306、削除 (Remove) ボタン1307をマウス装置106で操作することで行う。プログラム部品の追加、削除は、追加 (Add) ボタン1308、削除 (Remove) ボタン1309をマウス装置106で操作することで行う。候補プログラムの追加、削除、過去実現例からの参照は、それぞれ、追加 (Add) ボタン1311、削除 (Remove) ボタン1312、履歴 (History) ボタン1310をマウス装置106で操作することで行う。

【0155】また、候補部品の優先順位の変更は、上矢印ボタン1304、下矢印ボタン1305をマウス装置106で操作することで行う。

【0156】〔6. 初期アプリケーションの実行制御処理〕次に、初期構成のアプリケーションを実行する手順について、図2、図9、図11、図13、図14、図1

6、図17、図18を用いて説明する。図16は、本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いるアプリケーションプログラム実行制御処理ステップ208の詳細フローチャートであり、図17は、本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いるメッセージ配送プログラムのフローチャートであり、図18は、本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いるメッセージ配送プログラムの図17に続くフローチャートである。

【0157】図14に示した構成の初期構成のアプリケーションを実行するためには、図13に示すように、「販売管理サーバ」、「在庫管理」、「Stock Mgr」が選択された状態 (下線が付された状態) で、実行 (Run) ボタン1320をマウス装置106でクリックする。

【0158】その結果、図2に示した構成設定プログラムのステップ207の判定がYesとなり、ステップ208のアプリケーションプログラム実行制御処理が開始される。ステップ208の詳細については、図16に示してある。

【0159】ここで、図16のフローチャートについて、簡単に説明する。このフローは、入力の内容に応じて処理を起動するイベントフロー型となっている。

【0160】登録 (Register) ボタン1330 (図13) がマウス装置106によりクリックされると、ステップ1601において、入力がプログラム例の登録指示と判断され、ステップ1602において、指示されたメッセージを登録し、登録されたプログラム例を実行する処理が起動される。

【0161】削除 (Remove) ボタン1331 (図13) がマウス装置106によりクリックされると、ステップ1603において、入力がプログラム例の削除指示と判断され、ステップ1604において、指示されたメッセージに対するプログラム例を削除し、実行部品を実行する処理が起動される。

【0162】実行 (Run) ボタン1320がマウス装置106によりクリックされると、入力が実行指示と判断され、ステップ1606、1607を実行する。これらのステップについては、後述する。

【0163】実行中止 (Stop) ボタン1321 (図13) がマウス装置106によりクリックされると、ステップ1608において、入力が実行中止指示と判断され、ステップ1609において、アプリケーションプログラムの実行を中止する処理が起動される。

【0164】記録 (Record) ボタン1340 (図13) がマウス装置106によりクリックされると、ステップ1610において、入力が記録指示と判断され、ステップ1611において、構成記録処理が起動される。

【0165】なお、上述したように、ここでは、実行 (Run) ボタン1320がマウス装置によりクリックされるため、ステップ1606、1607について以下に詳述するとともに、ステップ1602、1604、1609、1611については、別途、[11. プログラム例の設定]、[8. 構成の記録]において詳述する。

【0166】ステップ1606において、アプリケーションプログラムの実行を開始する。

【0167】図14に示した販売管理サーバプログラム300は、接続プログラムであるサーバメイン (Server Main) プログラム310と、6つのコンポーネントデータ1410、1411、1412、1413、1414、1415から構成される。先に説明したように、在庫管理の部品機能をあらわすコンポーネントデータ (Stock Mgr Agent) 1410には、選択プログラム部品として「StockMgr」が内蔵されている。

【0168】図14で示したプログラムが作成され、これを実行開始する。

【0169】アプリケーションプログラムが、例えば、販売管理クライアントプログラム301 (図3) のように、画面を持つ場合には、ディスプレイ装置101 (図1) のアプリケーションプログラム画面111が表示される。

【0170】アプリケーションプログラムが、コンポーネントデータが含まれているからといっても変化なく、アプリケーションプログラム自体は部品から構成されているかのように動作する。これは、図9に示すメッセージ配送プログラム900が、初期構成では選択プログラム部品にメッセージ配送するためである。

【0171】ここで、図17のフローチャートを用いて、メッセージ配送プログラムの処理について簡単に説明する。このフローは、入力の内容に応じて処理を起動するイベントフロー型となっている。

【0172】プログラムの実行処理においては、メッセージ配送を行い、ステップ1701において、外部から受け取ったメッセージと、このメッセージの配送元をログテーブル914 (図9) に記録する。

【0173】ステップ1702は、プログラム例指定の判断ステップであり、Yesと判断されると、ステップ1703において、プログラム例指定時の処理が実行される。ステップ1704は、ジョーカー指定の判断ステップであり、Yesと判断されると、ステップ1705、1706において、ジョーカー指定の処理が実行される。なお、ステップ1703、1705、1706の詳細については、別途、[11. プログラム例の設定]、[10. ジョーカー部品の設定]において、説明する。

【0174】初期構成のアプリケーションプログラムの実行時には、ステップ1702、1704において、N

oと判断されるため、ステップ1707に至る。

【0175】ステップ1707において、メッセージが実行プログラム部品のインターフェースに一致するかが判断される。

【0176】初期構成のアプリケーションプログラムは、すでに、図3に示すようにして過去に実績のあるプログラムであるため、ステップ1707では、Yesと判断される。

【0177】ステップ1707の判断のためには、プログラム部品管理テーブルの一部であるメッセージ管理テーブル161 (図11) の情報にアクセスする。ここには、プログラム部品と、各プログラム部品が備えているメッセージインタフェースがすべて記録される。例えば、「Stock Mgr」プログラム部品は、プログラム部品管理識別子1102が「1」であり、これは「Add」、「Remove」、「Refer」などのメッセージインタフェース1103を有していることが図に示したテーブルから明らかである。このうち「Add」メッセージは、2つの引数を取り、第一引数の方はItem型、第二引数の型は整数型であることも示されている。従って、メッセージ管理テーブル161を参照して、メッセージが実行プログラム部品のインターフェースに一致するかが判断される。

【0178】ステップ1708において、実行プログラム部品にメッセージを配送し、メッセージ配送が成功であるか不成功であるかの結果を外部に出力する。外部出力としては、記憶装置へのダウンロードやプリンタへのプリントアウトがある。また、コンポーネントデータ130 (図9) のログテーブル914に、実行プログラム部品とメッセージ配送の結果を記録する。

【0179】なお、実行中にとびかうプログラム部品間のメッセージは、コンポーネントデータ130内のログテーブル914内に記録される。これもメッセージ配送プログラム900が行うものである。

【0180】なお、ステップ1707に続く図18に示すステップ1820以降の各ステップは、候補リスト中に候補として上げられたプログラム部品の取扱いに関するステップであり、この詳細内容は、別途、[9. 複数プログラム部品候補の設定]において説明する。

【0181】次に、図16のステップ1607に戻り、ステップ1607において、コンポーネントデータ130から通知されるメッセージ配送状況を画面表示する。即ち、ログテーブル914の内容は、逐次、構成設定プログラム123に報告されており、構成設定プログラム123は、逐次報告されてくるログテーブル914の内容を、構成設定画面110 (図13) に表示する。

【0182】構成設定画面110のMessageLog画面1325には、「Message」欄1322、「From」欄1323、「Response」欄1324の3欄が設けられている。「Message」欄1

322は、配送されたメッセージの内容を表しており、「From」欄1323は、メッセージの配送元を表しており、「Response」欄1324は、メッセージの配送状況を表している。

【0183】図13のMessageLog画面1325に示した5個のメッセージログは、以下のようなメッセージ配送状況を示している。第1行目は、在庫管理(Stock Mgr)プログラム部品が、在庫マスタデータベース(Stock DB)プログラム部品から「Locked」のメッセージを受取り、そのメッセージ配送の実行は、正常(OK)に行われたことを示している。第2行目は、在庫管理(Stock Mgr)プログラム部品が、在庫マスタデータベース(Stock DB)プログラム部品から「Released」のメッセージを受取り、そのメッセージ配送の実行は、正常(OK)に行われたことを示している。第3行目は、在庫管理(Stock Mgr)プログラム部品が、在庫画面(Stock Clt)プログラム部品から「Add」のメッセージを受取り、そのメッセージ配送の実行は、正常(OK)に行われたことを示している。第4行目は、在庫管理(Stock Mgr)プログラム部品が、在庫画面(Stock Clt)プログラム部品から「Retrieve」のメッセージを受取り、そのメッセージ配送の実行は、正常(OK)に行われたことを示している。第5行目は、在庫管理(Stock Mgr)プログラム部品が、在庫画面(Stock Clt)プログラム部品から「Refer」のメッセージを受取り、そのメッセージ配送の実行は、正常(OK)に行われたことを示している。

【0184】このメッセージログは、部品機能として選択された「Stock Mgr Agent」から報告された内容を示したものである。別の部品機能を選択することで、ログの表示を切り替えることもできる。

【0185】[7. プログラム部品の変更]次に、初期構成からプログラム部品を変更して代案を検討する手順について、図2、図4、図9、図10、図13、図19、図20を用いて説明する。図19は、本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法において表示する構成設定画面110の詳細な画面構成の構成図であり、図20は、本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いるコンポーネントデータの設定処理ステップ206の詳細フローチャートである。

【0186】ここでは、図4の初期構成から代案1への過程に示すように、管理部品の変更が検討されるものとする。即ち、初期構成では「Stock Mgr」をプログラム部品として採用したが、これが性能なり、機能なりで不十分であることが(実行を通して)評価されたと仮定している。

【0187】このとき、別のプログラム部品の導入が検

討される。「Stock Mgr」プログラム部品を改造して(プログラミングしなとして)使用することや、別の実現で使用されたプログラム部品を導入することが検討される。本実施の形態では、別の実現で使用されたプログラム部品お導入の検討に入ったと仮定する。ただし、部品の改造は、別のプログラム部品の開発とも解釈されるため、部品の改造は別の実現として別途開発され、プログラム部品管理テーブルに記録されたと考えれば、同様に扱うことができる。

【0188】ここでは、使用可能な別の在庫管理プログラム部品である「Stock Mgr_M」が存在していると考ええる。これは、プログラム部品管理テーブル160(図10)の識別子エントリ1001のフィールドに識別子「2」として記録されている。

【0189】「Stock Mgr」プログラム部品を「Stock Mgr_M」プログラム部品に変更するには、本実施の形態においては、部品名を選択し、これをキーボード装置を用いて変更することで行う。即ち、図13に示す構成設定画面110の状態において、コンポーネント(Component)サブ画面1302のID Name1316の「Stock Mgr」をマウス装置106でクリックして、「Stock Mgr」のプログラム部品を選択する(図13に示したように、「Stock Mgr」に下線が付された状態となる)。その後、キーボード装置105によって、「Stock Mgr」のプログラム部品名の後ろに、“_M”を追加する操作を行う。

【0190】上述したプログラム部品の変更操作によって、構成設定画面110の表示内容が図19に示すように変更される。なお、候補(Candidates)サブ画面1303、メッセージログ(Message Log)サブ画面1325の表示内容は、後程説明する処理の結果を表示しているものであり、その内容は、[9. 複数プログラム部品候補の設定]において後述する。

【0191】プログラム部品の変更操作が行われると、構成設定プログラム123(図1)は、図2に示すフローチャートのステップ202において、ユーザからの入力を受け付ける。さらに、ステップ205において、入力がプログラム部品の構成指定であると判断され、ステップ206に進む。ステップ206において、コンポーネントデータの設定処理が起動される。

【0192】コンポーネントデータの設定処理の詳細については、図20のフローチャートに示されている。

【0193】図20のステップ2001において、指示対象となっているプログラム部品に対応するコンポーネントデータを取得し、この取得したコンポーネントデータに対して処理を行う。

【0194】ステップ2002において、“_M”を追加する入力操作は、プログラム部品(「Stock M

10

20

30

40

50

g r _ M) の設定指示と判断され、S t o c k M g r A g e n t の実行プログラム部品が S t o c k M g r _ M に変更され、ステップ2003に進む。

【0195】ステップ2003において、コンポーネントデータ130 (図9) の選択プログラム履歴リスト913に実行プログラム部品で指定されたものを追加し、この実現名を「空」にする。即ち、以前選択されていたプログラム部品「S t o c k M g r」が、選択プログラム履歴リスト913に追加される。プログラム部品「S t o c k M g r」の識別子は、「1」であるため、プログラム部品識別子エントリ915のフィールドに「1」が追加される。但し、対応する実現名エントリ951は、「空」にする。

【0196】ステップ2004において、実行プログラム部品に指定プログラム部品を設定する。即ち、コンポーネントデータ130の選択プログラム部品エントリ910のフィールドに、「S t o c k M g r _ M」を設定する。

【0197】なお、ステップ2005は、ジョーカー部品に対する処理であり、新しいプログラム部品名称は、ジョーカーの名称ではないので、ステップ2005は実行されない。ジョーカー部品の場合については、[10. ジョーカー部品の設定] において後述する。

【0198】ここまでの操作で、プログラム部品の変更が完了する。後はこの構成でプログラムを実行し、評価することができる。これは、先に初期構成のプログラム実行で説明したものと同一手順で実行できる。ただし、ここでは、新しくプログラム部品が設定されたため、すべてのメッセージ呼び出しが成功するわけではない。成功か失敗かはメッセージログとして構成設定画面に表示される。なお、図19のメッセージログ (M e s s a g e L o g) サブ画面1325の表示内容は、後程説明するプログラム部品の候補を含めての処理の結果を表示しているものであり、その内容は、[9. 複数プログラム部品候補の設定] において後述する。

【0199】メッセージログの表示結果を評価の参考とすることができる。失敗の場合にはプログラムを改造し、全てが成功となるまでプログラミング作業が必要となる。本実施の形態ではこの作業については言及していない。本発明は、プログラム部品の変更 (選択) が容易に行えることに対してより効果を発揮する。

【0200】プログラム部品の変更を本発明によらずに行う場合には、ソースコードをテキストエディタで変更するなどの処理によって行うこととなる。これはコード編集、再コンパイル、実行といった手順となり、ここでのどのような変更がなされたか、変更後の評価は個別に管理する必要がある。

【0201】なお、図20に示したフローチャートにおいて、ステップ2006以降のステップは、プログラム部品の候補に対する処理であるため、後述する。

【0202】[8. 構成の記録] 次に新しい構成の記録について、図2, 図6, 図7, 図9, 図13, 図16, 図21, 図22, 図23を用いて説明する。図21は、本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いるアプリケーションプログラム実行制御処理ステップ208の中の構成記録処理ステップ1611の詳細フローチャートであり、図22は、本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いるプログラム構成管理テーブル150の構成図であり、図23は、本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いる実現管理テーブル151の構成図である。

【0203】プログラム部品の評価の結果として、記録に値するものが構成された場合、初期構成に対する代案として、新しい構成 (代案1) を設定することができる。これは、図13に示す構成設定画面110の記録指示ボタン1340をマウス装置106でクリックする操作で設定される。

【0204】記録指示の操作が行われると、構成設定プログラム123 (図1) は、図2に示すフローチャートのステップ202において、ユーザからの入力を受け付ける。さらに、ステップ207において、入力がアプリケーションプログラムの実行制御指示であると判断され、ステップ208に進む。ステップ208において、アプリケーションプログラムの実行制御処理が起動される。

【0205】さらに、図16のステップ1610において、記録指示の入力と判断され、ステップ1611の構成記録処理が起動される。構成記録処理の詳細については、図21を用いて説明する。

【0206】図21のステップ2101において、プログラム構成管理テーブル150 (図6) の実現名リストフィールドに、新たに付与した名称である実現名を記録する。即ち、図6に示すプログラム構成管理テーブル本体150の実現名リストエントリ603のフィールドに記録された識別子「1」に対応するフィールド (I m p l e m e n t a t i o n s フィールド) に、新たな名称の実現名「A2」を記録する。

【0207】この結果は、例えば、図22に示すようになる。即ち、図22に示すプログラム構成管理テーブル本体150には、新たな名称の実現名「A2」を記録される。なお、図22に示す状態では、実現名「A1」, 「A2」以外の実現名も記録されている状態を示しているが、これは、さらに、他の実現名も記録された状態を示したためである。

【0208】次に、ステップ2102において、実現管理テーブル151 (図7) に新しい名称の実現名を有するレコードを追加する。即ち、図7に示す状態では、実現名「A1」, 「A100」のレコードのみが記録され

ているが、このテーブルにA2を識別名称としてレコードを追加する。

【0209】この結果は、例えば、図23に示すようになる。即ち、図23に示す実現管理テーブル151には、識別子エントリ701のフィールドに識別子「A2」を追加する。

【0210】さらに、ステップ2103において、このレコードのコンポーネントデータリスト・フィールドに、全てのプログラム部品の機能単位のコンポーネントデータ識別子を記録する。即ち、図23に示すように、コンポーネントデータリストエントリ702のフィールドに記録された識別子「A2」に対応するフィールド（AIDsフィールド）に、すべてのプログラム部品単位のコンポーネントデータ識別子「1, 2, 3, 4, 5, 6」を記録する。

【0211】次に、ステップ2104において、このレコードの参照フィールドに、参照した実現の識別名称を記録する。即ち、図23に示すように、参照エントリ703のフィールドに記録された識別子「A2」に対応するフィールド（Prevフィールド）に、参照した実現の識別名称「A1」を記録する。

【0212】さらに、ステップ2105において、すべてのプログラム部品の機能単位に対応するコンポーネントデータ内の選択プログラム履歴の実現名フィールドのうち、「空」であるもののすべてにこの実現名を記録する。即ち、上述したステップ2003において、コンポーネントデータ130（図9）の選択プログラム履歴リスト913に実行プログラム部品で指定されたものを追加し、この実現名を「空」にしてあるが、この未記録の実現名に、実現名「A2」を記録する。

【0213】これに先に示した操作によって実現名が未定だった選択プログラム履歴（Stock Mgrなど）が、「A2」と記録される。この履歴記録は、後述する参照操作（[13. 以前の実現の参照]）の際に使用される。すなわち、実現を次々と変更した後で実現識別子「A2」を参照し、これを変更していくことが可能である。

【0214】なお、図23に示す状態では、実現名「A1」、「A100」、「A2」以外の実現名のレコードも記録されている状態を示しているが、これは、さらに、他の実現名のレコードも記録された状態を示したためである。

【0215】[9. 複数プログラム部品候補の設定] 次に、複数のプログラム部品候補の設定について、図2、図9、図11、図17、図18、図19、図20、図24を用いて説明する。図24は、本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法において構成された初期のアプリケーションプログラムである販売管理サーバプログラムの構成図である。

【0216】先の説明（[7. プログラム部品の変

更]）では、代案1の検討においてプログラム部品を取り替えることで実行評価を行っていた。本発明においては、これを複数の部品を同時に評価することも可能である。これは、同じ様な機能を有するプログラム部品をコンポーネントデータ内に並べて、メッセージ配送プログラムが入力されたメッセージによって実行可能なプログラム部品に振り分けることによって行っている。

【0217】プログラム部品の候補の追加は、図19に示す構成設定画面110において、コンポーネントサブ画面1316のプログラム部品機能を選択して（例えば、「Stock Mgr_M」を指定して）、追加（Add）ボタン1311をマウス装置106でクリックし、そこであらわれるダイアログで候補部品名称を指定することで行われる。

【0218】プログラム部品候補の追加操作が行われると、構成設定プログラム123（図1）は、図2に示すフローチャートのステップ202において、ユーザからの入力を受け付ける。さらに、ステップ205において、入力がプログラム部品の構成指定であると判断され、ステップ206に進む。ステップ206において、コンポーネントデータの設定処理が起動される。

【0219】コンポーネントデータの設定処理の詳細については、図20のフローチャートに示されている。

【0220】図20のステップ2001において、指示対象となっているプログラム部品に対応するコンポーネントデータを取得し、この取得したコンポーネントデータに対して処理を行う。

【0221】ステップ2006において、追加（Add）ボタン1311の入力操作は、プログラム部品候補の追加（Add）指示と判断されるので、ステップ2007が実行される。

【0222】ステップ2007において、指定された候補部品が、図9に示すコンポーネントデータ130の候補リスト911の第1番目のフィールド930に設定される。また、候補リスト911に設定されたプログラム部品候補は、図19に示す構成設定画面110の候補（Candidates）サブ画面1303に示す候補リストに表示される。図19に示す状態では、候補（Candidates）サブ画面1303には、候補部品として、「Stock Mgr_MM」が表示されている。

【0223】以上のようにして、プログラム部品の変更、追加が行われた結果、在庫管理エージェント（Stock Mgr Agent）1410の内部構成は、図24に例示したように、選択プログラム部品エントリには、「Stock Mgr_M」が格納され、候補リストエントリには、「Stock Mgr_MM」が格納され、選択プログラム履歴エントリには、「Stock Mgr」が格納され、プログラム例エントリは、「空」の状態となっている。

【0224】以上の操作を繰り返すことで、候補部品は、候補リスト911のフィールド931、932に順次追加される。また、その追加結果は、候補(Candidates)サブ画面1303に表示される。

【0225】候補リスト内の候補部品の順序は、リスト内の実行優先順位を意味している。実行優先順位を変更するには、図19の上矢印ボタン1304、下矢印ボタン1305をマウス装置106でクリックすることで行える。

【0226】即ち、上矢印ボタン1304、下矢印ボタン1305が操作されると、図20のステップ2010において、プログラム部品候補の上(下)移動指示と判断され、ステップ2011が実行される。

【0227】ステップ2011において、指定されたプログラム部品を、候補リスト911のフィールド内で上位(下位)に移動する。移動結果は、候補(Candidates)サブ画面1303に表示される。

【0228】また、候補(Candidates)サブ画面1303に表示されたプログラム部品候補を削除する場合には、削除するプログラム部品を候補(Candidates)サブ画面1303において指定し、削除(Remove)ボタン1312を操作することにより行われる。

【0229】削除(Remove)ボタン1312が操作されると、図20のステップ2008において、プログラム部品候補の削除と判断され、ステップ2009が実行される。

【0230】ステップ2009において、指定されたプログラム部品を、候補リスト911のフィールドから削除する。上位のフィールドが削除された時は、下位のフィールドのプログラム部品が上位に移動する。

【0231】候補の追加設定や、候補順位の変更が行われて、図19の構成設定画面110に表示される候補リスト設定が行われた場合について考える。ここで、プログラム実行を指示した際、コンポーネントデータ内のメッセージ配送プログラムは、以下に示す処理を実行する。

【0232】コンポーネントサブ画面1302に表示され、プログラム実行の指定された「Stock Mgr_M」(実行プログラム部品)が処理可能なメッセージは、ステップ1707、1708(図17)によって、「Stock Mgr_M」のコンポーネントデータ130(図9)に配送し、結果などをログテーブル914に記憶する。

【0233】そうでない場合には、ステップ1707で「No」であるので、ステップ1820に進む。

【0234】ステップ1820において、候補プログラム部品リストに未処理部品が存在するか否かを判断する。即ち、候補リスト911(図9)の各フィールド930、931、932を順位に従ってチェックする。こ

こでは、未処理のプログラム部品候補として、「Stock Mgr_MM」が登録されているので、ステップ1821に進む。

【0235】ステップ1821において、候補部品リストの最上位の未処理部品を選択し、処理済みとする。即ち、プログラム部品「Stock Mgr_MM」を選択する。

【0236】さらに、ステップ1822において、メッセージが、選択された候補部品のインターフェースに一致するか否かが判断される。即ち、メッセージ管理テーブル161(図11)を参照することによって、実行可能なメッセージかどうかを判断する。メッセージのインターフェースが一致する場合には、ステップ1823に進む。

【0237】ステップ1823において、選択候補部品にメッセージを配送し、結果を外部出力するとともに、ログテーブルに選択候補部品名と結果を記録する。即ち、選択されているプログラム部品候補「Stock Mgr_MM」にメッセージを配送し、結果をポート出力する。また、ログテーブル914(図9)に選択候補部品名と結果を記録する。

【0238】なお、図18のステップ1822において、メッセージが、選択された候補部品のインターフェースに一致しない場合には、ステップ1820に戻って、全ての未処理部品について同様の処理を行う。

【0239】ステップ1820において、インターフェースの一致するプログラム部品が候補リストにない場合には、ステップ1824に進む。

【0240】ステップ1824において、いずれの候補部品のメッセージインターフェースが一致しない場合には、配送部品なしと記録する。その結果、構成設定画面110のメッセージログ1325には「Error」と表示する。

【0241】ここで、図19の構成設定画面110のメッセージログ(Message Log)サブ画面1325は、以上のようにして、候補部品についてもメッセージ配送を実行した結果の一例を表示している。

【0242】図19のMessage Log画面1325に示した5個のメッセージログは、以下のようなメッセージ配送状況を示している。第1行目は、在庫管理(Stock Mgr_M)プログラム部品が、在庫マスタデータベース(Stock DB)プログラム部品から「Locked」のメッセージを受取り、そのメッセージ配送の実行は、正常(OK)に行われたことを示している。

【0243】第2行目は、在庫管理(Stock Mgr_M)プログラム部品が、在庫マスタデータベース(Stock DB)プログラム部品から「Released」のメッセージを受取り、そのメッセージ配送の実行は、正常(OK)に行われたことを示している。

【0244】第3行目は、在庫管理 (Stock Mgr_M) プログラム部品が、在庫画面 (Stock Clt) プログラム部品から「Add」のメッセージを受取り、そのメッセージ配送の実行は、正常 (OK) に行われたことを示している。

【0245】第4行目は、在庫管理 (Stock Mgr_M) プログラム部品が、在庫画面 (Stock Clt) プログラム部品から「Retrieve」のメッセージを受取ろうとしたが、「Retrieve」のメッセージが、在庫管理 (Stock Mgr_MM) プログラム部品のインターフェースに一致しないため、候補リストから在庫管理 (Stock Mgr_MM) プログラム部品を選択し、在庫管理 (Stock Mgr_MM) プログラム部品が、在庫画面 (Stock Clt) プログラム部品から「Retrieve」のメッセージを受取り、そのメッセージ配送の実行は、正常 (OK) に行われたことを示している。

【0246】第5行目は、在庫管理 (Stock Mgr_MM) プログラム部品が、在庫画面 (Stock Clt) プログラム部品から「Refer」のメッセージを受取ろうとしたが、インターフェースが一致せず、それ以外にインターフェースの一致するプログラム部品がプログラム部品候補として存在しなかったことを、“Error” で表示している。

【0247】以上の処理により、実行時のメッセージ配送の状況は、図19に示すメッセージログサブ画面1325の、特に、応答 (Response) 欄1324でモニタすることができる。

【0248】図19からは、「Stock Mgr_M」は、上の3つのメッセージは処理可能であるが、下の2つは処理できないことがわかる。また、プログラム部品候補「Stock Mgr_MM」によって、4番めの処理が実行可能なこともわかる。

【0249】これらの結果から、「Stock Mgr_M」は、「Stock Mgr_MM」の一部を参照してプログラミングすることで、構成をより満足するプログラムとすることができるとなどが評価できる。

【0250】また、複数候補を並行評価することができることで、1つずつ評価するより迅速な評価作業となることも期待される。

【0251】[10. ジョーカー部品の設定] 次に、ジョーカー部品の設定について、図2、図9、図13、図17、図20、図25を用いて説明する。図25は、本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法において表示する構成設定画面110の詳細な画面構成の構成図である。

【0252】ここで、最初に、ジョーカー部品について説明する。

【0253】プログラム部品の候補として、場合によっては何を設定すればよいか、わからない場合も考えられ

る。プログラム部品管理テーブルを検索し、メッセージ管理テーブルを参照することでそれらしい部品を見つけることはできるが、これはデータベース問い合わせを行うことと等しい。問い合わせの手順を踏まず、実行を通じてインターフェースが一致するプログラム部品を検索できるようにするのが、ジョーカー部品である。

【0254】ジョーカー部品が設定された場合には、候補リストのプログラム部品要素を検索し、実行時には、配送されたメッセージに応答可能な (形式が一致する) プログラム部品に、メッセージ配送する。

【0255】ジョーカー部品により、適すると思われるいくつかのプログラム部品候補を提示されれば、構成作成業者にとって、検索されたプログラム部品の内容より呼び出しインターフェースなどから適用可能なプログラム部品を絞り込むことができる。

【0256】ジョーカー部品は、名前“*”で表される。これを実行プログラムに指定することができるものとする。

【0257】ジョーカー部品を指定するには、本実施の形態においては、部品名を選択し、これをキーボード装置を用いて変更することで行う。即ち、図13に示す構成設定画面110の状態において、コンポーネント (Components) サブ画面1302のID Name1316の「Stock Mgr」をマウス装置106でクリックして、「Stock Mgr」のプログラム部品を選択する (図13に示したように、「Stock Mgr」に下線が付された状態となる)。その後、キーボード装置105によって、“*”に変更する。

【0258】上述したジョーカー部品の指定操作によって、構成設定画面110の表示内容が図25に示すように変更される。なお、メッセージログ (Message Log) サブ画面1325の表示内容は、後程説明する処理の結果を表示しているものであり、その内容は後述する。

【0259】ジョーカー部品の指定操作が行われると、構成設定プログラム123 (図1) は、図2に示すフローチャートのステップ202において、ユーザからの入力を受け付ける。さらに、ステップ205において、入力がプログラム部品の構成指定であると判断され、ステップ206に進む。ステップ206において、コンポーネントデータの設定処理が起動される。

【0260】コンポーネントデータの設定処理の詳細については、図20のフローチャートに示されている。

【0261】図20のステップ2001において、指示対象となっているプログラム部品に対応するコンポーネントデータを取得し、この取得したコンポーネントデータに対して処理を行う。

【0262】ステップ2002において、“*”の入力操作は、ジョーカー部品の指定と判断され、ステップ2005に進む。

【0263】ステップ2005において、対応するコンポーネントデータ130のメッセージ配送プログラム900（図9）にジョーカー指定を通知する。

【0264】図17に示すメッセージ配送プログラム900のステップ1704において、メッセージ配送プログラム900は、ジョーカー指定されていることを判断し、ステップ1705に進む。

【0265】ステップ1705において、メッセージを条件としてプログラム部品管理テーブル群のメッセージ管理テーブル161を検索し、そのメッセージにインタフェースが一致するプログラム部品を選択する。

【0266】次に、ステップ1706において、自動的に、インタフェースが一致するプログラム部品を主記憶装置102にロードし、これにメッセージ送信し、結果をポート出力する。また、ログテーブル914（図9）には、部品名と結果を記録する。

【0267】以上の処理によって、データベース問い合わせを行うことなく、実行時の文脈で適用可能なプログラム部品を検索することが可能である。場合によっては検索結果が正しくシステムに適用可能な場合もあり、このときは検索と組み込みを一度に済ませることも可能である。

【0268】ジョーカ部品の実行例が、図25に示されている。ここでは、在庫管理（Stock Mgr Agent）プログラム部品にジョーカー部品を設定している。ここで、実行を指示すると、メッセージログサブ画面1325にメッセージの配送結果が示されるように、プログラム部品「Stock Mgr」に5個のメッセージに対応しており、配送可能であることが示されている。

【0269】「Stock Mgr」は、初期構成で実行プログラムとして指定したプログラム部品であり、自明な検索例であるが、他の例でも動作することは明らかである。

【0270】データベース検索のためには、あらかじめどのようなメッセージ処理が必要とされているかを調査する手順も含まれる。この調査のためには、全システムを精査する必要がある。また、メッセージも動的に（システム要求に従って）変更されることも考えられる。これらを考慮すれば、実行時文脈の検索は、単に手順の省略というにとどまらず、一時的にシステムの動作を保証するための手段としても有効である。

【0271】本実施の形態によれば、ジョーカー部品が設定された場合には、候補リストのプログラム部品要素を検索し、適すると思われるいくつかのプログラム部品候補を提示できるので、構成作成作業にとって、検索されたプログラム部品の内容より呼び出しインタフェースなどから適用可能なプログラム部品を容易に絞り込むことができる。

【0272】[11. プログラム例の設定] 次に、プロ

グラム例の設定について、図9、図16、図17、図26を用いて説明する。図26は、本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法において表示する構成設定画面110の詳細な画面構成の構成図である。

【0273】最初に、プログラム例について説明する。

【0274】これまでにプログラム部品として実現されておらず、かつ、プログラミング作業の工数がおおきく見積もられる場合などには、これまでに提供した手段によってプログラムを続行させることができない。こういった場合に、本発明では、エージェントであるコンポーネントデータが、プログラム部品を経由せず、直接メッセージに応答する手段を提供する。これがコンポーネントデータ130（図9）内に設定されるプログラム例である。プログラム例エントリ912には、メッセージに応じたユーザの設定するプログラム例を格納される。

【0275】プログラム例として、他のプログラムから呼び出された場合に期待される処理を、プログラム例エントリ912に記述することが可能である。このエントリに登録されており、メッセージ形式が一致する場合には、メッセージ配送プログラムは、このプログラム例にメッセージを配送する。即ち、プログラム部品のうち、未完成部分の仮実行コードをプログラム例として記述することで、アプリケーション全体の動作を確認評価できる。

【0276】図26は、プログラム例（Example）サブ画面1332にプログラム例を設定した例を示している。ここでは、在庫管理部機能に要求される「Add」メッセージに、プログラム例を設定している。このプログラムは、プログラム例（Example）サブ画面1332内にキーボード装置105を用いて入力され、登録（Register）ボタン1330をマウス装置106でクリックすることで登録される。

【0277】登録（Register）ボタン1330がクリックされると、図2に示した構成設定プログラムのステップ207の判定がYesとなり、ステップ208のアプリケーションプログラム実行制御処理が開始される。ステップ208の詳細については、図16に示してある。

【0278】図16のステップ1601において、入力プログラム例の登録指示と判断され、ステップ1602において、指示されたメッセージを登録し、登録されたプログラム例を実行する処理が起動される。ここでは、指示されたメッセージ「Add」に対して、図26のプログラム例（Example）サブ画面1332内に示すプログラム例の実行をエージェントに指示している。

【0279】プログラムの実行時には、Addメッセージが入力された場合に、メッセージ配送プログラム900は、図17のステップ1702において、プログラム

例指定がなされていると判断し、ステップ1703に進む。

【0280】ステップ1703において、プログラム例を実行し、結果をポート出力する。また、コンポーネントデータ130（図9）のログテーブル914には、プログラム例の実行とその結果を記録される。

【0281】ログテーブル914の内容は、図26のMessageLog画面1325に表示される。プログラム例を実行した時には、[Example]と表示される。即ち、メッセージログの中で、第3行目に示されるように、在庫画面（Stock Clt）プログラム部品から「Add」のメッセージを、プログラム例[Example]が受取り、実行していることを示している。

【0282】次に、プログラム例の削除について説明する。

【0283】削除（Remove）ボタン1312（図13）をクリックされると、構成設定プログラム123（図1）は、図2に示すフローチャートのステップ202において、ユーザからの入力を受け付ける。さらに、ステップ207において、入力がアプリケーションプログラムの実行制御指示と判断され、ステップ208に進む。

【0284】ステップ208において、アプリケーションプログラムの実行制御処理が起動される。

【0285】アプリケーションプログラムの実行制御処理を示す図16のステップ1603において、プログラム例の削除（Remove）ということで、選択されているプログラムにプログラム例を削除する処理を実行する。

【0286】ステップ1604において、指示されたメッセージに対するプログラム例を削除する。

【0287】以上によって、実行制御の削除処理が実行される。

【0288】プログラム例設定は、工数が大きい場合以外にも、プログラム部品の副作用（データベース登録など）を起こしたくない場合や、初期段階でメッセージ整合性だけをとってシステム動作を確認する場合などにも有効に作用する。

【0289】本実施の形態によれば、他のプログラムから呼び出された場合に期待される処理、即ち、プログラム部品のうち、未完成部分の仮実行コードをプログラム例として、プログラム例エントリ912に記述することにより、メッセージ形式が一致する場合には、メッセージ配送プログラムは、このプログラム例にメッセージを配送するので、アプリケーション全体の動作を確認評価できる。

【0290】以上説明したように、[7. プログラム部品の変更]、[9. 複数候補の設定]、[10. ジョーカー部品]、[11. プログラム例]の各手段によつ

て、代案を次々と設定し、実現を繰り返すことができる。即ち、各プログラム部品を”実際に組み込んで評価する”ことが可能なことが明かである。本実施の形態のはじめに述べた代案1から代案4までは連続的に実現していくことができる。

【0291】[12. 部品機能構成の変更] 次に、プログラム部品機能単位の構成を変更する場合について、図7、図8、図13、図27を用いて説明する。図27は、本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いる機能構成の設定処理ステップ204の詳細フローチャートである。

【0292】部品機能単位の追加、削除は、システム開発中に行われることも多いものである。本実施の形態では先に、クライアント側にサーバデータベースのローカルキャッシュの役割を果たすデータベースを設定する例を述べたが、これが機能単位の変更にあたる。

【0293】機能単位の追加は、構成設定画面で部品追加を指示することで行う。即ち、構成設定画面110

（図13）の追加（Add）ボタン1308をマウス装置106でクリックすることで、部品機能追加処理が起動される。

【0294】追加（Add）ボタン1308をクリックされると、構成設定プログラム123（図1）は、図2に示すフローチャートのステップ202において、ユーザからの入力を受け付ける。さらに、ステップ203において、入力がアプリケーションプログラムの機能構成指定と判断され、ステップ204に進む。

【0295】ステップ204において、機能構成の設定処理が起動される。機能構成の設定処理の詳細については、図27のフローチャートに示されている。

【0296】図27のステップ2705において、プログラム機能単位の追加（Add）ということで、選択されているプログラムにプログラム部品機能を追加する処理を実行する。

【0297】ステップ2706において、機能単位に識別子を与え、対応するコンポーネントデータを作成する。機能単位の名称はダイアログ内にキーボードで入力する。これによって、構成設定処理は新しい機能単位に対応するコンポーネントデータを作成する。

【0298】次に、ステップ2707において、実現管理テーブル151（図7）の作業中の実現名を有するレコードのコンポーネントデータリストエントリ702のフィールドに機能単位の識別子を追加する。また、プログラム構成管理テーブルにも新しいコンポーネントデータに対応した記録がなされる。

【0299】さらに、ステップ2708において、コンポーネントデータ管理テーブル152（図8）に、機能単位の識別子と機能単位名（コンポーネント名）を有するレコードを記録する。

【0300】以上によって、部品機能追加処理が実行さ

れる。

【0301】また、プログラム機能単位の削除も先の説明の逆の手順によって行われる。

【0302】削除(Remove)ボタン1309(図13)をクリックされると、構成設定プログラム123(図1)は、図2に示すフローチャートのステップ202において、ユーザからの入力を受け付ける。さらに、ステップ203において、入力がアプリケーションプログラムの機能構成指定と判断され、ステップ204に進む。

【0303】ステップ204において、機能構成の設定処理が起動される。

【0304】機能構成の設定処理を示す図27のステップ2710において、プログラム機能単位の削除(Remove)ということで、選択されているプログラムにプログラム部品機能を削除する処理を実行する。

【0305】ステップ2711において、実現管理テーブル151(図7)の作業中の実現名を有するレコードのコンポーネントデータリストエントリ702の機能単位の識別子を削除する。

【0306】次に、ステップ2712において、コンポーネントデータ管理テーブル152(図8)に、機能単位の識別子を有するレコードを削除する。

【0307】以上によって、部品機能削除処理が実行される。但し、コンポーネントデータは後で参照されることがあるため、削除を実行しないなどの保全処置をとっている。

【0308】ここで説明したプログラム機能単位の追加によって、代案0から代案5を作成することが可能である。但し、直前の実現を参照していないため、以前の実現を参照する手段が必要となる。これを次に説明する。

【0309】[13. 以前の実現の参照] 次に、過去に作成した実現を参照して新しい実現とする履歴参照場合について、図2、図9、図19、図20、図23、図28、図29、図30を用いて説明する。図28は、本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いるコンポーネントデータの設定処理ステップ206の中のプログラム代行履歴参照処理のフローチャートであり、図29は、本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いるプログラム代行履歴参照処理の履歴画面の構成図であり、図30は、本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いるプログラム代行履歴参照処理時のコンポーネントデータの構成図である。

【0310】アプリケーションプログラムの構成作成にあたっては、変更履歴を管理し、いつでも任意の組み合わせ構成に戻せることが必要である。そこで、選択プログラム履歴リストを採用している。選択プログラム履歴リストの要素には、プログラム部品レコードと、実現名

称の組合わせが記録される。ここで、実現名称は組み合わせの実現例を識別するためのものである。任意の時点に実現名称が入力されることで、すべてのコンポーネントデータの履歴リストを検索し、組み合わせを再現することができる。

【0311】本実施の形態では、以前の実現の参照は、図19に示す構成設定画面110の履歴(History)ボタン1310をマウス装置106でクリックすることで行う。

10 【0312】履歴参照指示の操作が行われると、構成設定プログラム123(図1)は、図2に示すフローチャートのステップ202において、ユーザからの入力を受け付ける。さらに、ステップ205において、入力がコンポーネントデータ設定処理の指示であると判断され、ステップ206に進む。ステップ206において、コンポーネントデータ設定処理が起動される。

【0313】さらに、図20のステップ2012において、履歴参照指示の入力と判断され、ステップ2013のプログラム代行履歴参照処理が起動される。プログラム代行履歴参照処理の詳細については、図28を用いて説明する。

【0314】図28のステップ2801において、指示されたコンポーネントデータの実行プログラム部品の履歴が、ダイアログに表示される。

【0315】図29を用いて、ダイアログに表示される履歴画面について説明する。履歴画面には、指示された指示されたプログラム部品機能単位の実行プログラム履歴が表示される。ここでは、図19のプログラム部品分類を表示する分類(Classification)画面1315の中の「在庫管理」を選択して、履歴(History)ボタン1310がクリックされたものとする。

【0316】履歴画面の構成は、プログラム部品機能単位である「在庫管理」の右側に、コンポーネントデータサブ画面2901が表示される。また、その下には、履歴サブ画面2902が表示される。コンポーネントデータサブ画面2901及び履歴サブ画面2902への表示は、コンポーネントデータ130(図9)内の選択プログラム履歴リスト913を参照することで実行される。

40 【0317】図30は、コンポーネントデータである在庫管理エージェント(StockMgr Agent)1410の内部構成を示している。在庫管理エージェント(StockMgr Agent)1410の選択プログラム部品エントリには、「StockMgr_M」が格納され、候補リストエントリには、「StockMgr_MM」が格納され、選択プログラム履歴リストエントリには、「StockMgr」、「StockMgr_M」、「StockMgr_MM」に対する実現名が格納され、プログラム例エントリは、
50 「空」の状態となっている。

【0318】従って、コンポーネントデータサブ画面2901には、参照したコンポーネントデータ名の「Stock Mgr」が表示され、履歴サブ画面2902には、在庫管理エージェント（Stock Mgr Agent）1410の選択プログラム履歴リストのフィールドに記録された過去の実現であるプログラム部品名が表示されるが、この内容については、後述する。

【0319】さらに、ステップ2802において、履歴画面の右側に、アプリケーション実現の履歴を（有向）グラフによって表示する。グラフ構造は、ノードとノード間の遷移を示す方向付きアークで表される。図29に示すグラフの例は、図4に示した代案履歴を表したものであり、これは、実現管理テーブル151（図23）の参照エントリ703のPrevフィールドをたどること

で得ることができる。なお、図23は、代案6を検討時の実現管理テーブル151の例となっている。図4に示す「初期構成」が、実現名「A1」に対応し、「代案1」が実現名「A2」に対応し、以下同様にして、「代案6」が実現名「A6」に対応している。

【0320】ここで、例えば、図23に示されるように、実現名「A2」のPrevフィールドは実現名「A1」であり、したがって、「A1」のノードから「A2」のノードに向かうアークを引くことができる。同様にして、実現名「A2」から実現名「A7」に至るツリー構造を表示できる。

【0321】次に、ステップ2805において、グラフのノードの一つが選択されたか否かが判断される。ノードの選択は、グラフのノードの一つをマウス装置106でクリックすることで行われる。選択されると、ステップ2806が実行される。

【0322】ステップ2806において、選択されたノードの実現時の選択プログラム履歴が履歴サブ画面2902に表示される。例えば、図29に示す構成で、「A1」のノードをマウス装置106でクリックすることで、「A1」で設定された実行プログラム部品だけを履歴サブ画面2902内に表示することができる。これは、選択プログラム履歴リストのフィールドに記録された過去の実現名に「A1」が含まれているものだけを履歴サブ画面2902内に表示する処理によって実現している。ここでは、「Stock Mgr」、「Stock Mgr_M」が表示されている。

【0323】ここで、ステップ2803に戻って、履歴サブ画面2902内に表示された一つのプログラム部品が選択されると、ステップ2804が実行される。

【0324】ステップ2804において、選択された実行プログラム部品を変更候補とする。

【0325】次に、部品の変更について説明する。

【0326】ステップ2807において、部品の変更指示が行われたか否かが判断される。部品の変更指示は、図27に示すダイアログ内で、変更（Change）ボ

タン2904をマウス装置106でクリックすることで行われる。

【0327】次に、ステップ2808において、変更部品をエージェント内の実行プログラム部品としてプログラム部品の変更が行われ、ダイアログを消去する。

【0328】また、取消（Cancel）ボタン2905がマウス装置106でクリックされた場合には、ステップ2909において、キャンセル指示が行われたと判断され、ステップ2910において、変更がキャンセルされ、ダイアログが消去される。

【0329】この変更によって、図4に示す代案1から代案5につなげることも可能であるし、代案6で代案4の実現を参照することも可能となる。

【0330】以上説明した本実施の形態により、部品を取り替えながら（選択しながら）プログラムを実行させ、そのなかで評価を行うことが可能となる。

【0331】本実施の形態によれば、選択プログラム履歴リストを採用し、任意の時点に選択プログラム履歴リストの実現名称が入力されることで、すべてのコンポーネントデータの履歴リストを検索し、組み合わせを再現することができるので、アプリケーションプログラムの構成作成にあたっては、変更履歴を管理し、いつでも任意の組み合わせ構成に戻せることが可能となる。

【0332】[14. アプリケーションプログラムの変更] 次に、アプリケーションプログラムを変更する場合について、図2、図6、図13、図27を用いて説明する。

【0333】アプリケーションプログラムの追加は、構成設定画面でプログラム追加を指示することで行う。即ち、構成設定画面110（図13）の追加（Add）ボタン1306をマウス装置106でクリックすることで、アプリケーションプログラム追加処理が起動される。

【0334】追加（Add）ボタン1306をクリックされると、構成設定プログラム123（図1）は、図2に示すフローチャートのステップ202において、ユーザからの入力を受け付ける。さらに、ステップ203において、入力アプリケーションプログラムの機能構成指定と判断され、ステップ204に進む。

【0335】ステップ204において、機能構成の設定処理が起動される。機能構成の設定処理の詳細については、図27のフローチャートに示されている。

【0336】図27のステップ2701において、アプリケーションプログラムの追加（Add）ということで、アプリケーションプログラムを追加する処理を実行する。

【0337】ステップ2702において、プログラム構成管理テーブル150（図6）に入力されたプログラム名を持ち、実現リストを空にしたレコードを追加する。

【0338】以上によって、アプリケーションプログラ

ム追加処理が実行される。

【0339】また、アプリケーションプログラムの削除の場合、削除 (Remove) ボタン 1307 (図 13) をクリックされると、構成設定プログラム 123

(図 1) は、図 2 に示すフローチャートのステップ 202 において、ユーザからの入力を受け付ける。さらに、ステップ 203 において、入力がアプリケーションプログラムの機能構成指定と判断され、ステップ 204 に進む。

【0340】ステップ 204 において、機能構成の設定処理が起動される。 10

【0341】機能構成の設定処理を示す図 27 のステップ 2703 において、アプリケーションプログラムの削除 (Remove) ということで、アプリケーションプログラムを削除する処理を実行する。

【0342】ステップ 2704 において、プログラム構成管理テーブル 150 (図 6) から入力されたプログラム名を持つレコードを削除する。

【0343】以上によって、アプリケーションプログラム削除処理が実行される。 20

【0344】以上説明したように、本実施の形態によれば、多数のプログラム部品を組み合わせるアプリケーションを構成する場合に、引き続き構成プログラム部品の選択 (評価) を実際に動作させながら行うことが可能となる。

【0345】従って、データベース支援による方法に比較して、実際に動作させることで機能、性能、デザインをより詳細に検討可能となる。

【0346】また、ここでは、コンポーネントデータ内にプログラム部品を内蔵させる方式をとったことで、プログラムを直接変更させたり、対話処理系の実行中にプログラム部品を交換するような方式に比較して、評価の枠組みとしての独立性を高くしたものである。 30

【0347】この独立性は、複数部品を候補としたり、ジョーカー部品を設定したり、プログラム例を設定できる自由度となって現れている。個々の機能は、従来の対話処理系などでも可能だが、それぞれを簡単な操作で、かつ互いに影響を及ぼさずに組み込み可能な点が本実施の形態の特徴であり、効果となっている。

【0348】また、部品を選択する場合、作成するアプリケーションでどのようなメッセージが送信されるかを確認した後で選択することが可能である。どの場面、どのメッセージが使用されたかの情報は、部品選択の補助情報とすることができる。これはジョーカー部品のように、とりあえず動作するプログラム構成とする場合に、特に有効に作用する。ジョーカー部品とプログラム例だけを使用して、システムの仕様を確認する作業なども可能である。 40

【0349】また、選択された系列を後戻りすること、別の系列のものを参照することも、実現系列とその実現 50

時に検討した内容を含めて可能となっている。これは 1 つの実現で検討した内容の再検討を行う場合などに有効である。

【0350】総論として、本実施の形態により期待される効果としては、机上の計算による評価にとどまらず、実環境での評価を効率的に行うことができることから、より要求に即した開発が可能となることである。

【0351】

【発明の効果】本発明によれば、アプリケーションプログラムの構成作成支援方法において、プログラム部品を組み合わせる過程で、次々に実行し、評価を行うことを可能となる。

【図面の簡単な説明】

【図 1】本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法において使用するハードウェアと、主なソフトウェア要素を示したシステム構成図である。

【図 2】本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いる構成設定プログラムの概要を示すフローチャートである。

【図 3】本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法を説明するための過去に作成されたアプリケーションプログラムの構成図である。

【図 4】本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法による変更の過程を示す説明図である。

【図 5】本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法により最終的に作成されたアプリケーションプログラムの構成図である。

【図 6】本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いるプログラム構成管理テーブル 150 の構成図である。

【図 7】本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いる実現管理テーブル 151 の構成図である。

【図 8】本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いるコンポーネントデータ管理テーブル 152 の構成図である。

【図 9】本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いるコンポーネントテーブル 130 の構成図である。

【図 10】本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いるプログラム部品管理テーブル 160 の構成図である。

【図 11】本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いるメッセージ管理テーブル 161 の構成図である。

【図 12】本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いる初期設定処理

ステップ 201 の詳細フローチャートである。

【図 13】本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法において表示する構成設定画面 110 の詳細な画面構成の構成図である。

【図 14】本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法において構成された初期のアプリケーションプログラムである販売管理サーバプログラムの構成図である。

【図 15】本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法において構成された初期のアプリケーションプログラムである販売管理クライアントプログラムの構成図である。

【図 16】本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いるアプリケーションプログラム実行制御処理ステップ 208 の詳細フローチャートである。

【図 17】本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いるメッセージ配送プログラムのフローチャートである。

【図 18】本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いるメッセージ配送プログラムの図 17 に続くフローチャートである。

【図 19】本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法において表示する構成設定画面 110 の詳細な画面構成の構成図である。

【図 20】本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いるコンポーネントデータの設定処理ステップ 206 の詳細フローチャートである。

【図 21】本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いるアプリケーションプログラム実行制御処理ステップ 208 の中の構成記録処理ステップ 1611 の詳細フローチャートである。

【図 22】本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いるプログラム構成管理テーブル 150 の構成図である。

【図 23】本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いる実現管理テーブル 151 の構成図である。

【図 24】本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法において構成された初期のアプリケーションプログラムである販売管理サーバ

プログラムの構成図である。

【図 25】本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法において表示する構成設定画面 110 の詳細な画面構成の構成図である。

【図 26】本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法において表示する構成設定画面 110 の詳細な画面構成の構成図である。

【図 27】本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いる機能構成の設定処理ステップ 204 の詳細フローチャートである。

【図 28】本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いるコンポーネントデータの設定処理ステップ 206 の中のプログラム代行履歴参照処理のフローチャートである。

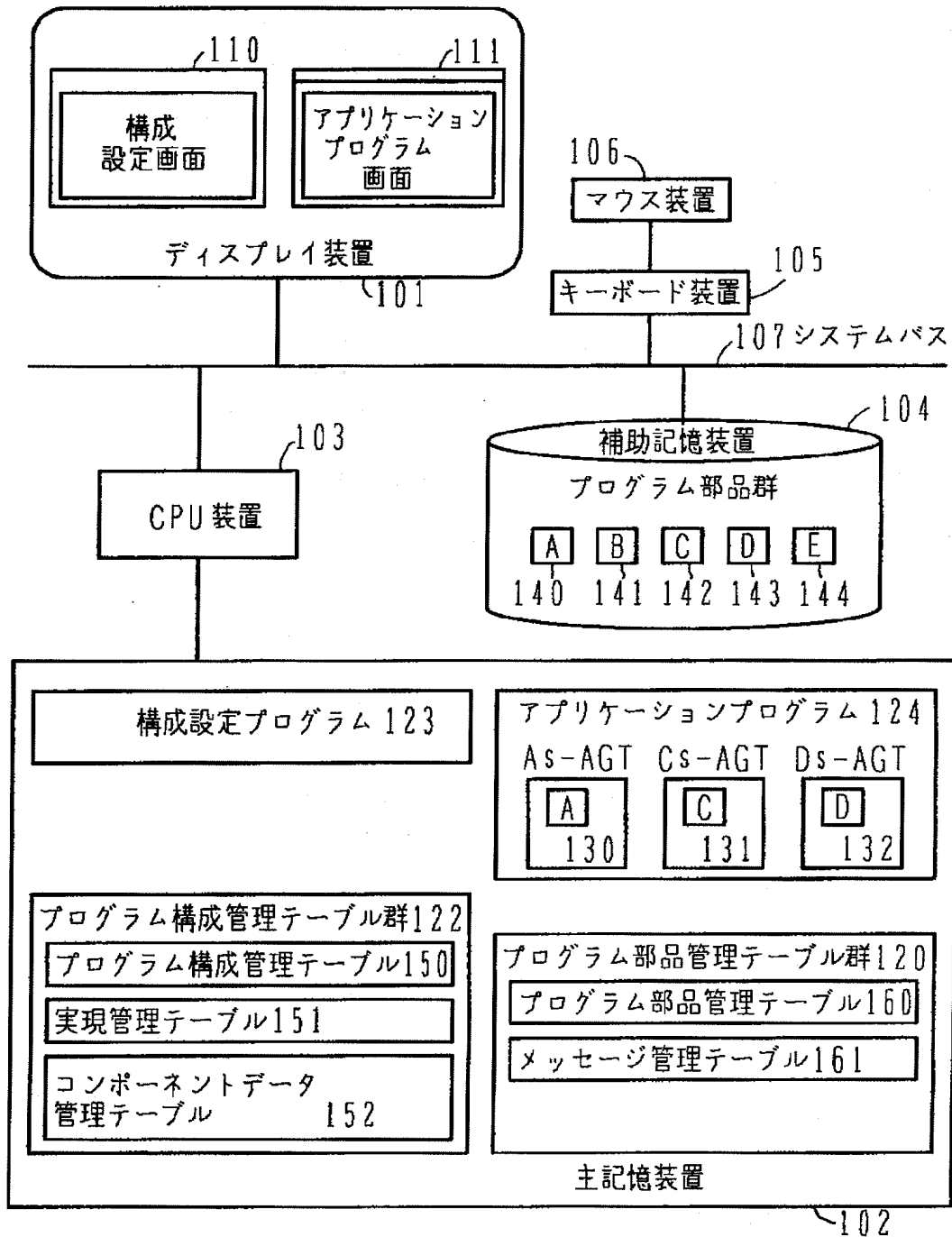
【図 29】本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いるプログラム代行履歴参照処理の履歴画面の構成図である。

【図 30】本発明の一実施の形態によるアプリケーションプログラムの構成作成支援方法に用いるプログラム代行履歴参照処理時のコンポーネントデータの構成図である。

【符号の説明】

- 101…ディスプレイ装置
- 102…主記憶装置
- 103…CPU装置
- 104…補助記憶装置
- 105…キーボード装置
- 106…マウス装置
- 107…システムバス
- 110…インタフェース画面
- 111…アプリケーションプログラム画面
- 120…プログラム部品管理テーブル群
- 123…構成設定プログラム
- 122…プログラム構成管理テーブル群
- 124…アプリケーションプログラム
- 130, 131, 132…コンポーネントデータ
- 140, 141, 142, 143, 144…プログラム部品
- 150…プログラム構成管理テーブル本体
- 151…実現管理テーブル
- 152…コンポーネントデータ管理テーブル
- 160…プログラム部品管理テーブル本体
- 161…メッセージ管理テーブル

【図1】



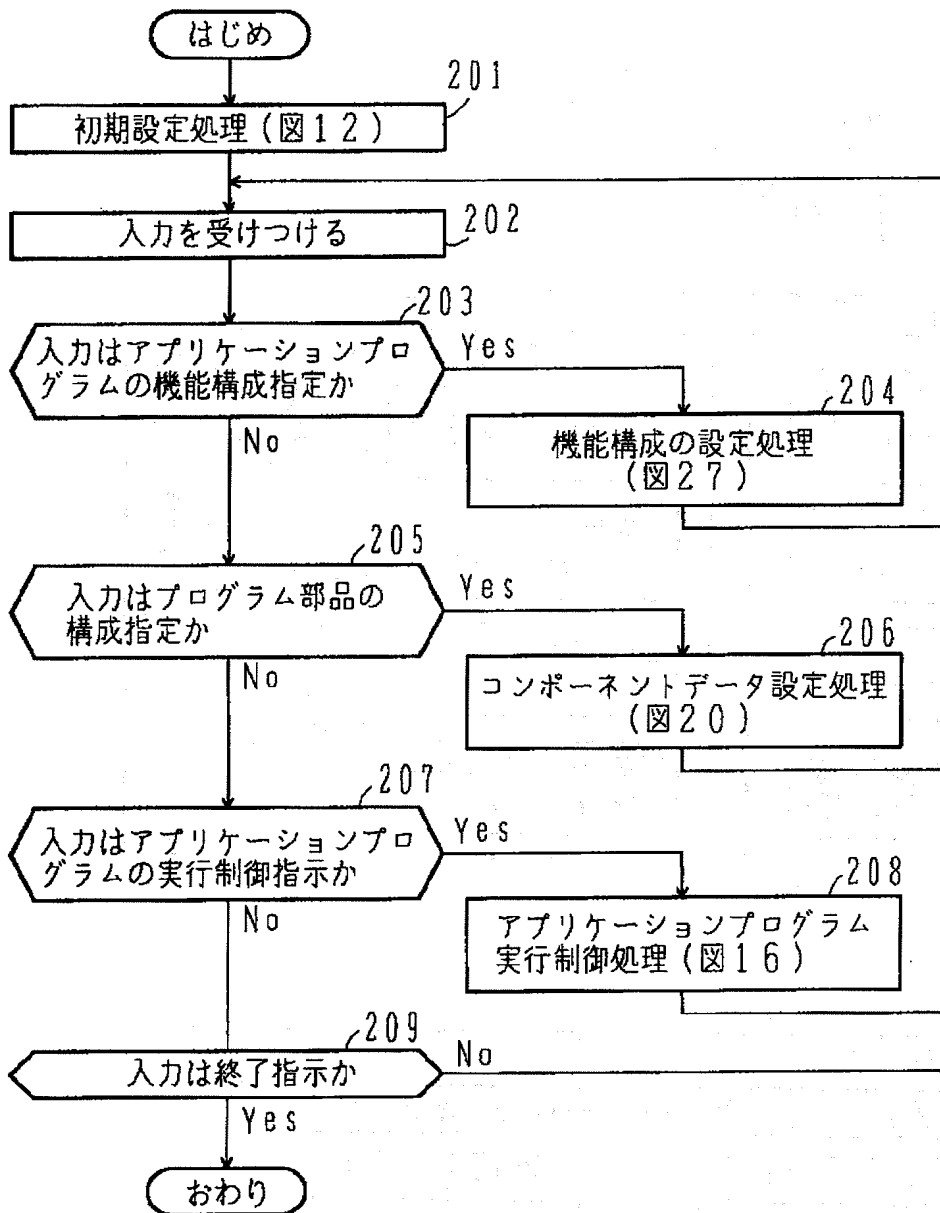
【図6】

150		
識別子	プログラム名	実現名リスト
1	販売管理サーバ	A1
2	販売管理クライアント	A100

【図7】

151			
識別子	コンポーネント データリスト	参照	コンポーネント 外リスト
A1	1,2,3,4,5,6	-	Main0
A100	7,8,9	-	Main100

【図 2】



【図 8】

152

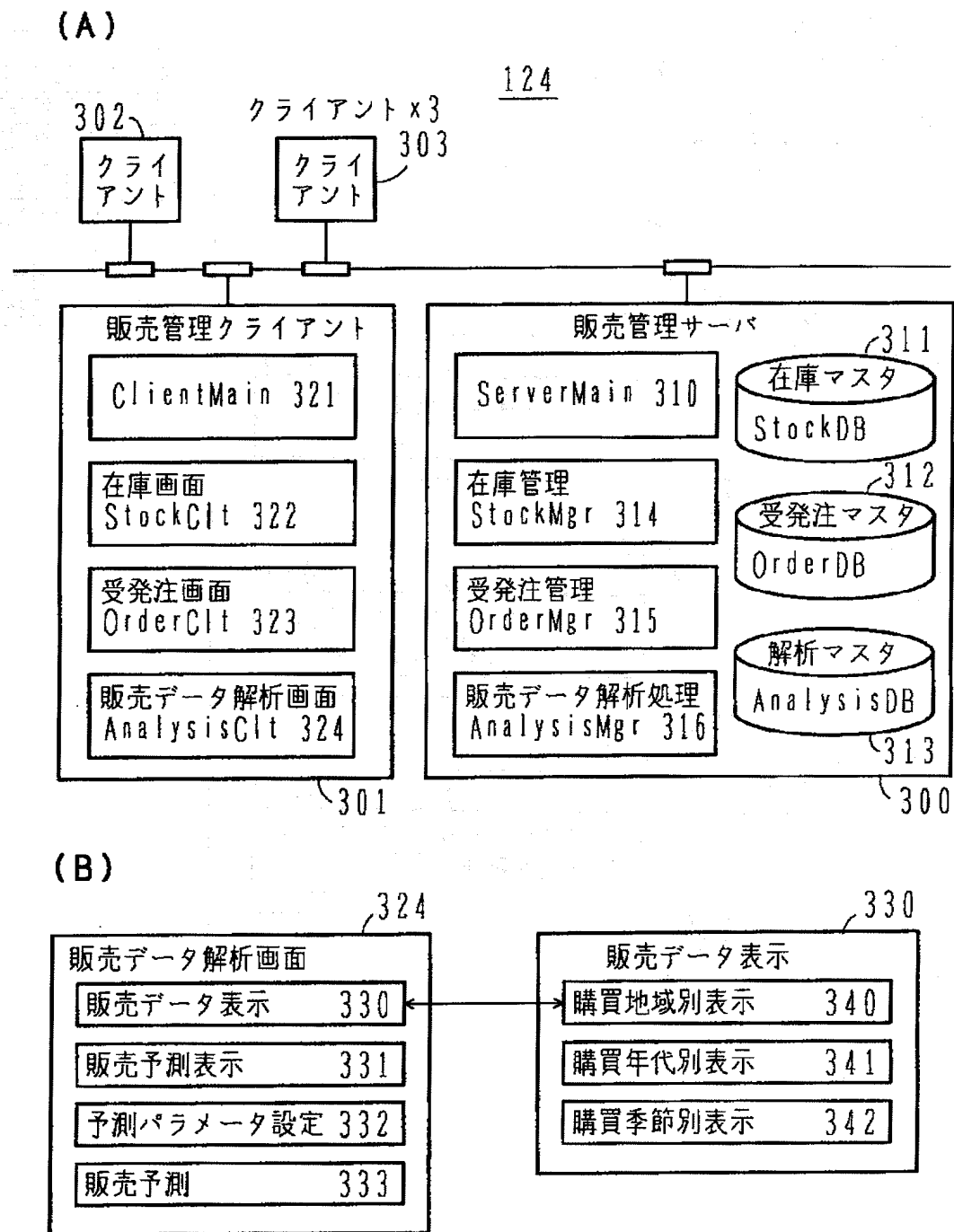
識別子	コンポーネント名
1	StockMgrAgent
2	OrderMgrAgent
3	AnalysisMgrAgent
4	StockDBAgent
5	OrderDBAgent
6	AnalysisDBAgent
7	StockClitAgent
8	OrderClitAgent
9	AnalysisClitAgent

【図 22】

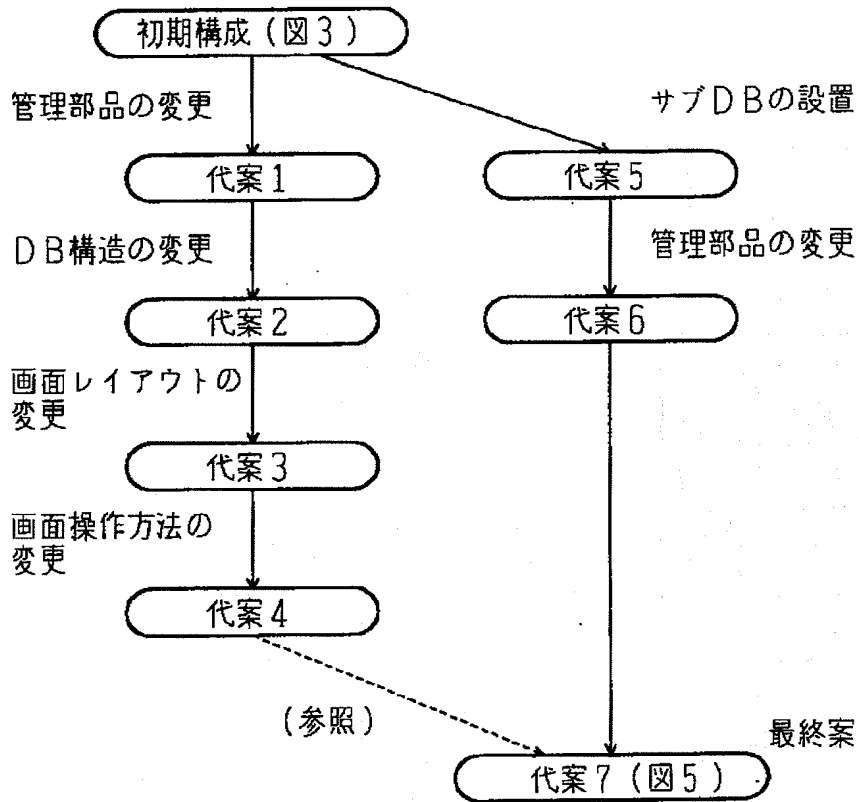
150

識別子	プログラム名	実現名リスト
1	販売管理サーバ	A1,A2,A3,A4,A5,A6
2	販売管理クライアント	A100,A101,A102,A103,A104,A105

【図3】



【図4】



【図30】

StockMgrAgent 1410

選択プログラム部品

StockMgr_M

候補リスト

StockMgr_MM

プログラム例

空

選択プログラム履歴リスト

StockMgr	A1
StockMgr_M	A1,A2,A3,A5,A6
StockMgr_MM	A2

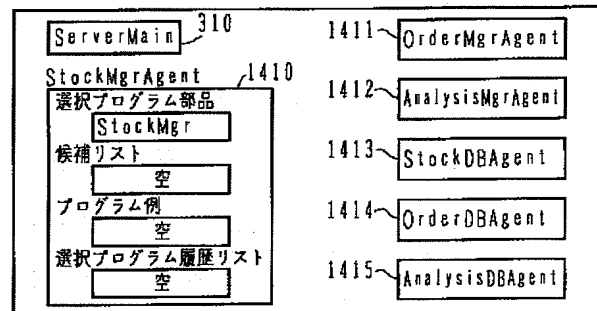
【図10】

160

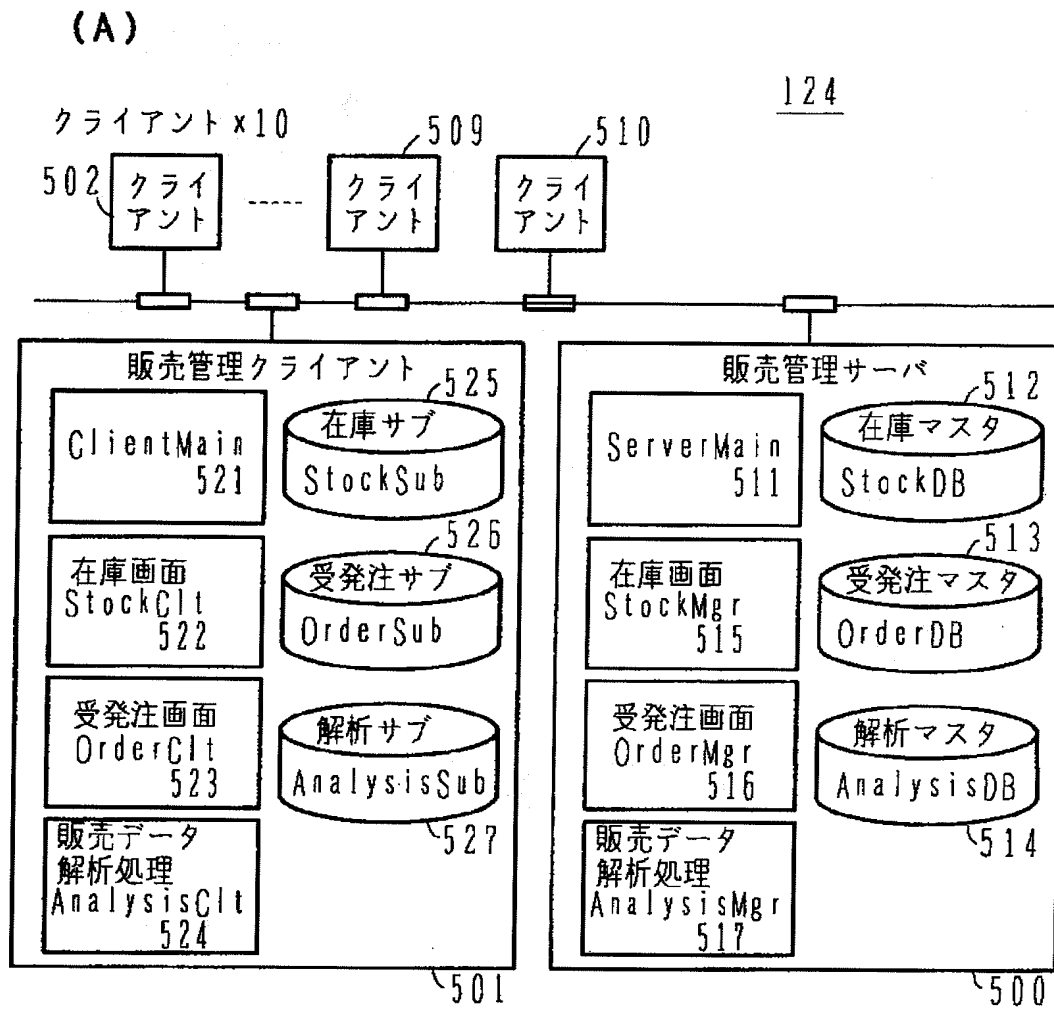
1001 識別子	1002 プログラム部分類	1003 プログラム名	1004 接続実績リスト
1	在庫管理	StockMgr	95,96,520,522
2	在庫管理	StockMgr_M	95,97,98,522
3	在庫管理	AtockMgr_MM	95,98,523
...			
13	受発注管理	OrderMgr	124,125,527
14	受発注管理	OrderMgr_M	125,530
...			
28	販売データ解析管理	AnalysisMgr	127,535
...			
95	在庫管理マスタ	StockDB	1,2,3
124	受発注管理マスタ	OrderDB	13
127	販売データ解析マスタ	AnalysisDB	28
520	在庫管理画面	StockClt	1
527	受発注画面	OrderClt	13,15
535	販売データ解析画面	AnalysisClt	28
...			
Main0	販売管理サーバ制御	ServerMain	1,13,28,95,124,127
...			
Main100	販売管理クライアント制御	ClientMain	520,527,535

【図14】

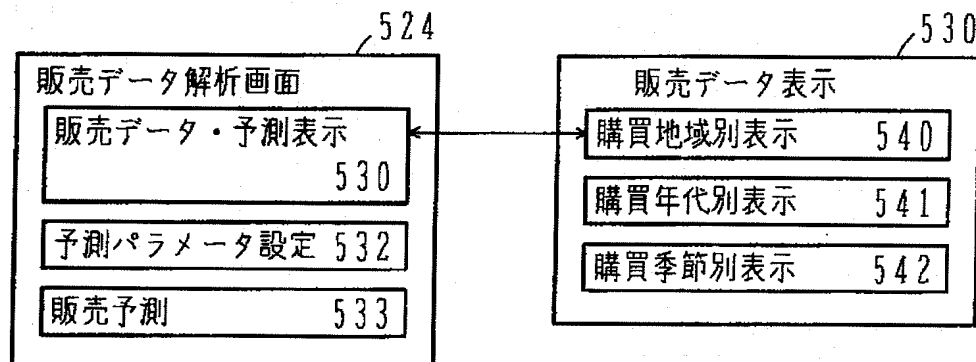
300



【図5】

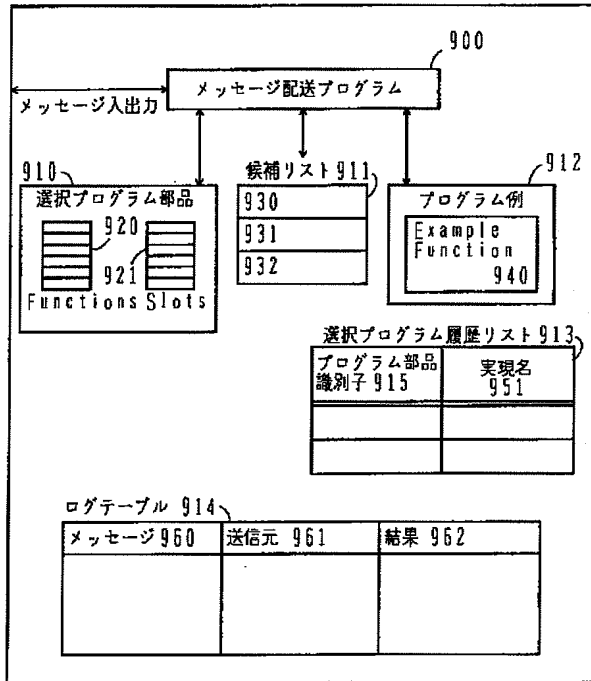


(B)



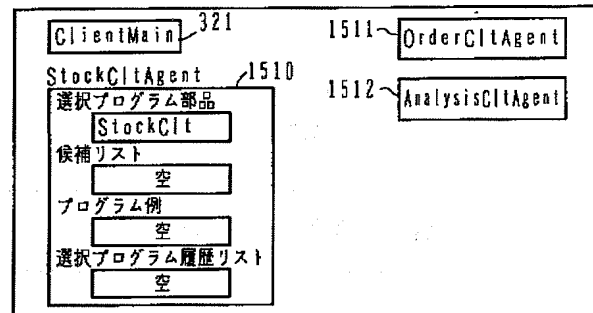
【図9】

130



【図15】

301



【図23】

151

701 識別子	702 コンポーネント データリスト	703 参照	704 コンポーネント 外リスト
A1	1,2,3,4,5,6	-	Main0
A100	7,8,9	-	Main100
A2	1,2,3,4,5,6	A1	Main0
A101	7,8,9	A100	Main100
A3	1,2,3,4,5,6	A2	Main0
A102	7,8,9	A101	Main100
A4	1,2,3,4,5,6	A3	Main0
A103	7,8,9	A102	Main100
A5	1,2,3,4,5,6	A1	Main0
A104	7,8,9,10,11,12	A100	Main100
A6	1,2,3,4,5,6	A5	Main0
A105	7,8,9,10,11,12	A104	Main100

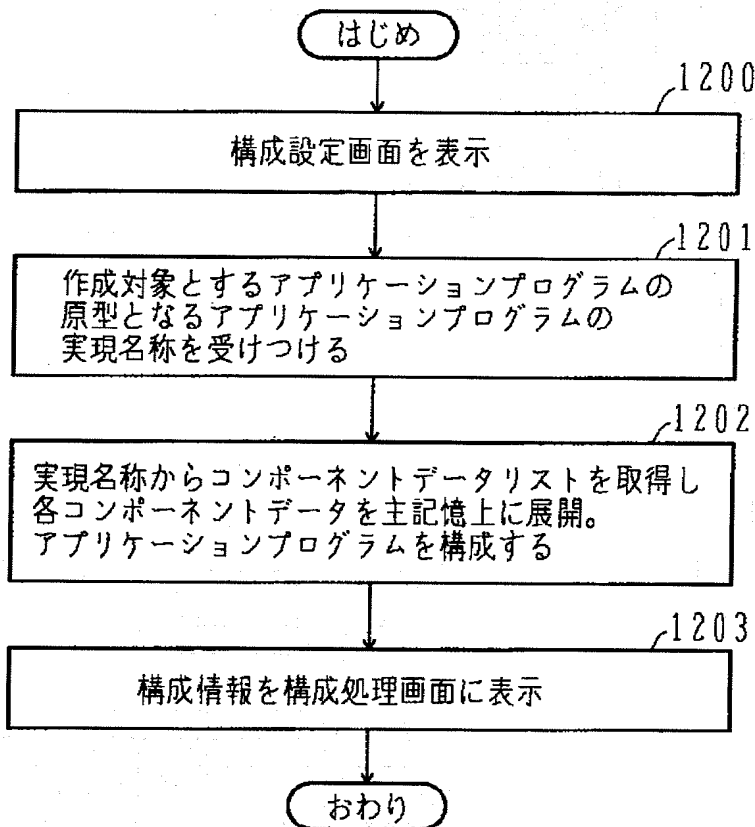
【図11】

161

1101 識別子	1102 プログラム 部品識別子	1103 メッセージ名	1104 ArgCount	1105 Argument1	1106 Argument2	1107 Argument3
1	1	Add	2	Item	int	
2	1	Remove	2	Item	int	
3	1	Refer	2	Item	int&	
---	---					
9	2	Remove	2	Item	int	
10	2	Refer	2	Item	int&	
---	---					

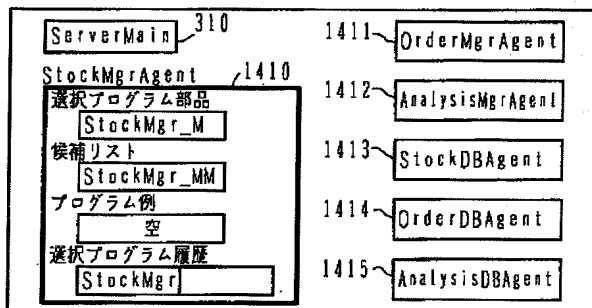
【図12】

201

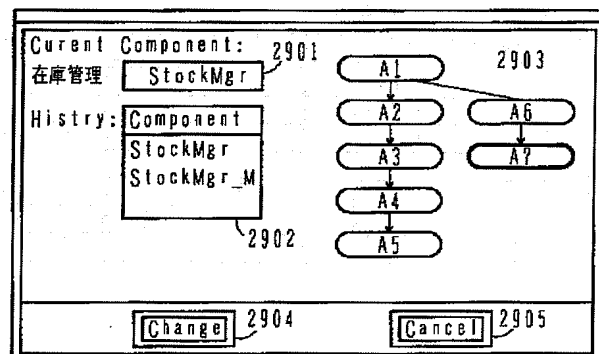


【図24】

300



【図29】



【図13】

110

110

Program 1301 Components 1315 Candidates 1304 1305

販売管理サーバ
販売管理クライアント

Classification 1316

在庫管理 StockMgr
受発注管理 OrderMgr
販売データ解析 AnalysisMgr
在庫マスタ StockDB
受発注マスタ OrderDB
解析マスタ AnalysisDB

1306 1307 1308 1309 1302 1310 1311 1312

1320 1321

Run Stop

Message Log 1322 1323 1324

Message	From	Response
void Locked(void)	StockDB	StockMgr::OK
void Released(void)	StockDB	StockMgr::OK
void Add(Item, Count)	StockClient	StockMgr::OK
void Retrieve(Item, Count)	StockClient	StockMgr::OK
void Refer(Item, Count&, Date&, arrival)	StockClient	StockMgr::OK

1325

Example: 1332 1330 Register Remove 1331

Record 1340 1341 Exit

【図19】

110

110

Program 1301 Components 1315 Candidates 1304 1305

販売管理サーバ
販売管理クライアント

Classification 1316

在庫管理 StockMgr_M
受発注管理 OrderMgr
販売データ解析 AnalysisMgr
在庫マスタ StockDB
受発注マスタ OrderDB
解析マスタ AnalysisDB

1306 1307 1308 1309 1302 1310 1311 1312

1320 1321

Run Stop

Message Log 1322 1323 1324

Message	From	Response
void Locked(void)	StockDB	StockMgr_M::OK
void Released(void)	StockDB	StockMgr_M::OK
void Add(Item, Count)	StockClient	StockMgr_M::OK
void Retrieve(Item, Count)	StockClient	StockMgr_M::OK
void Refer(Item, Count&, Date&, arrival)	StockClient	Error

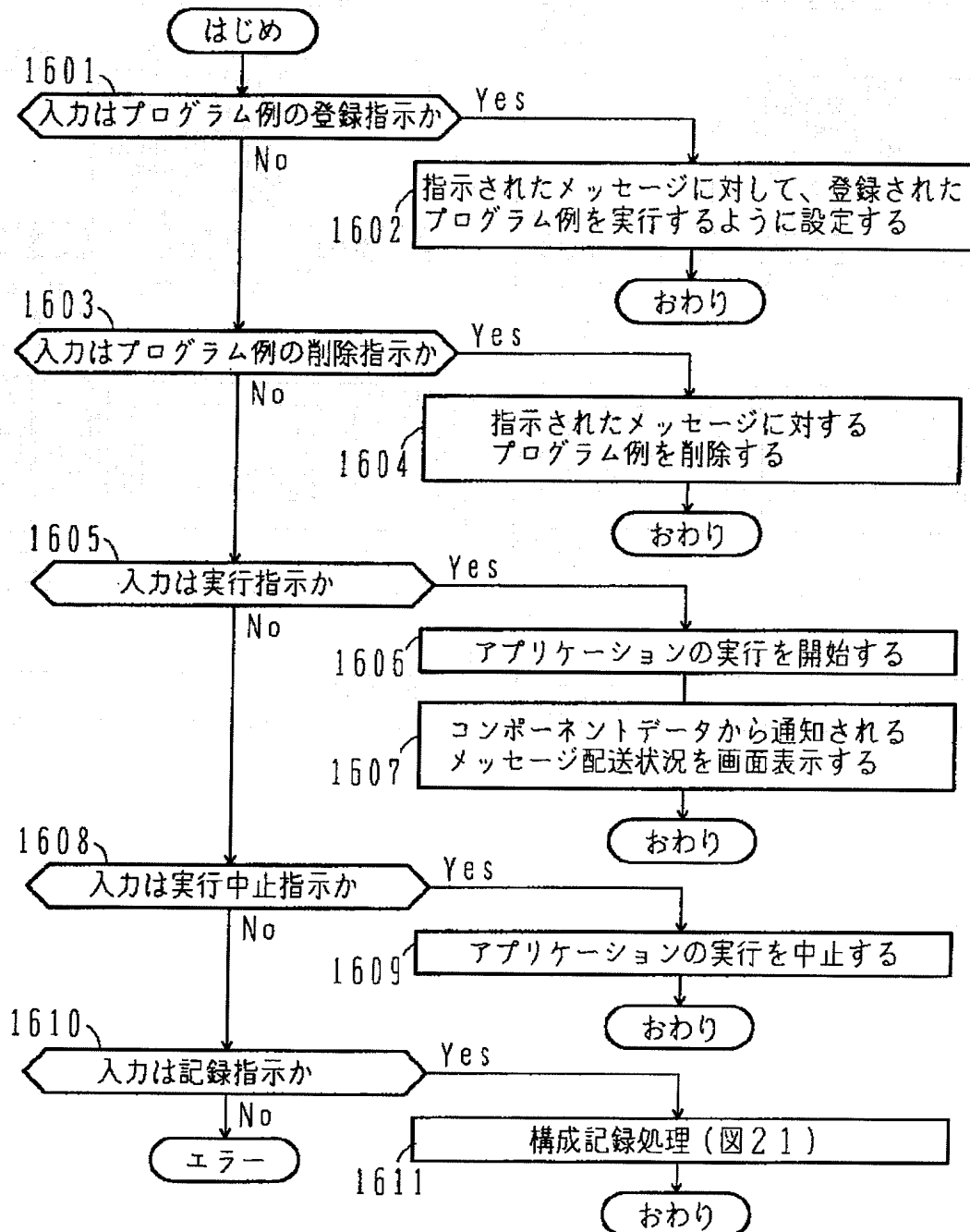
1325

Example: 1332 1330 Register Remove 1331

Record 1340 1341 Exit

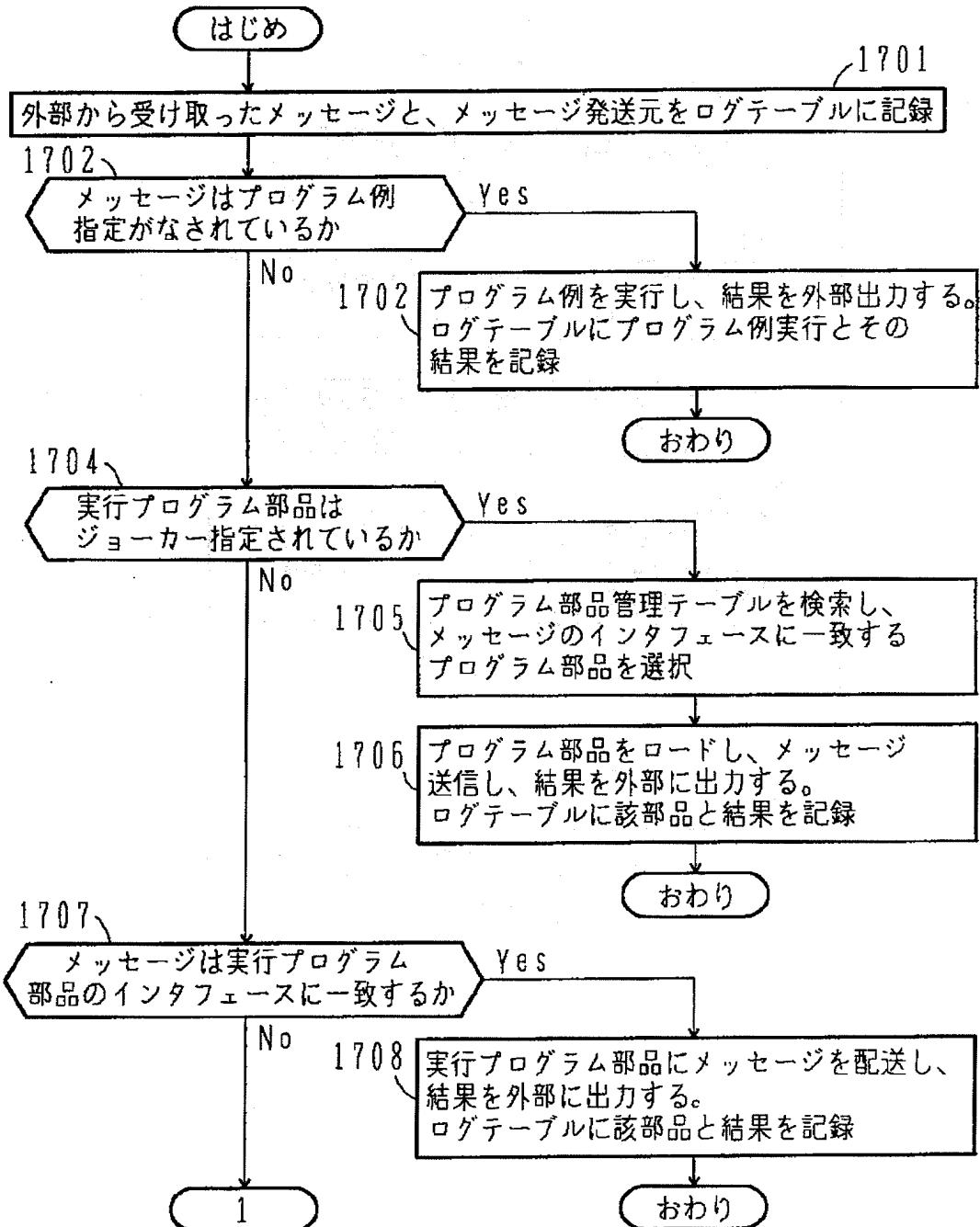
【図16】

208

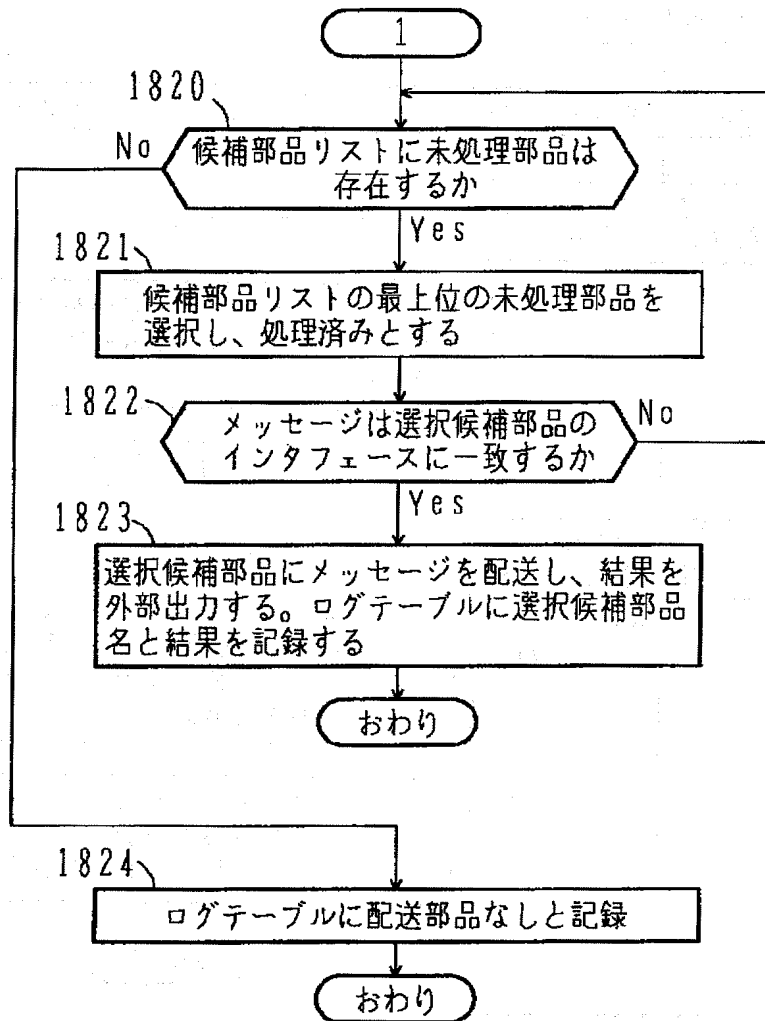


【図17】

900

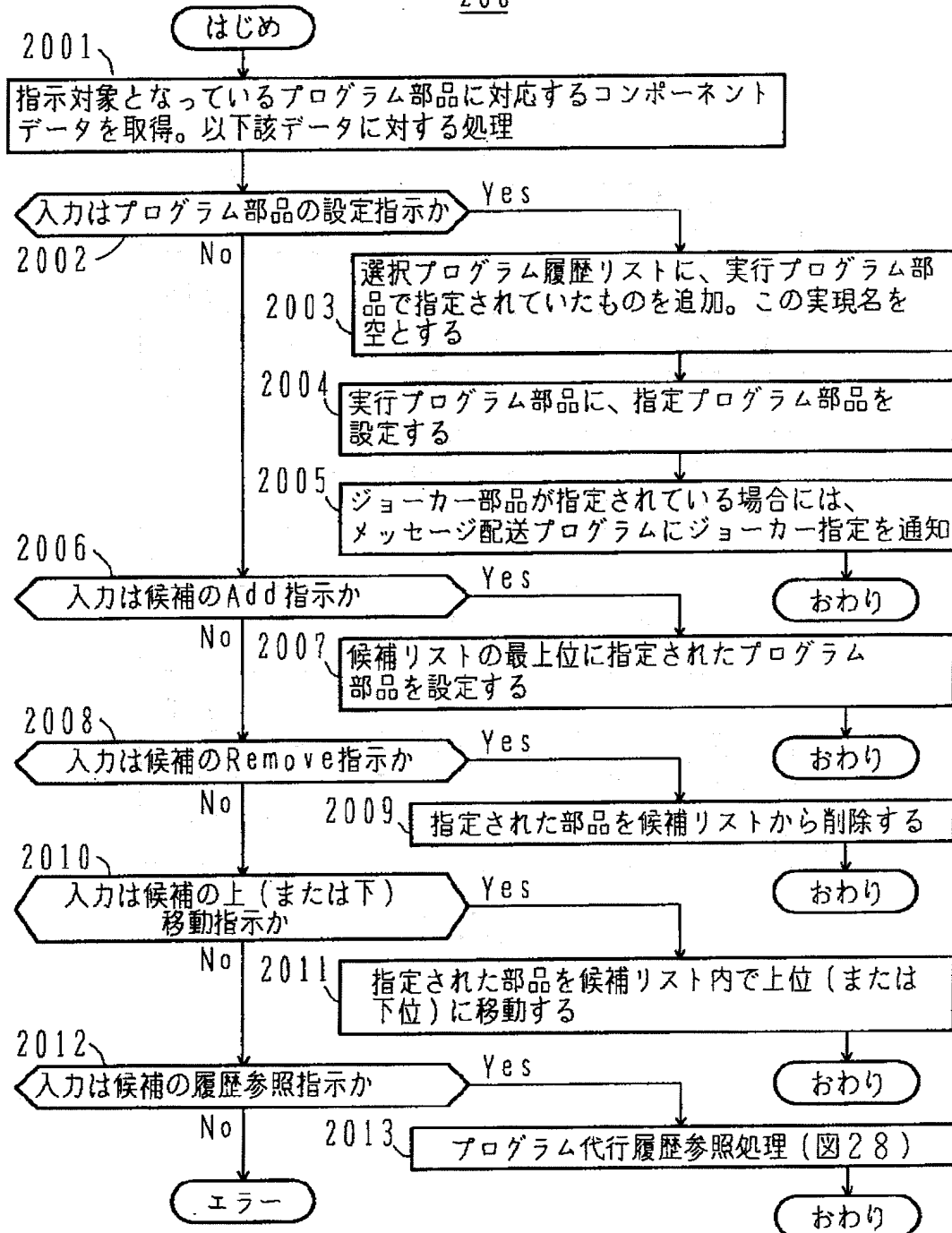


【図18】

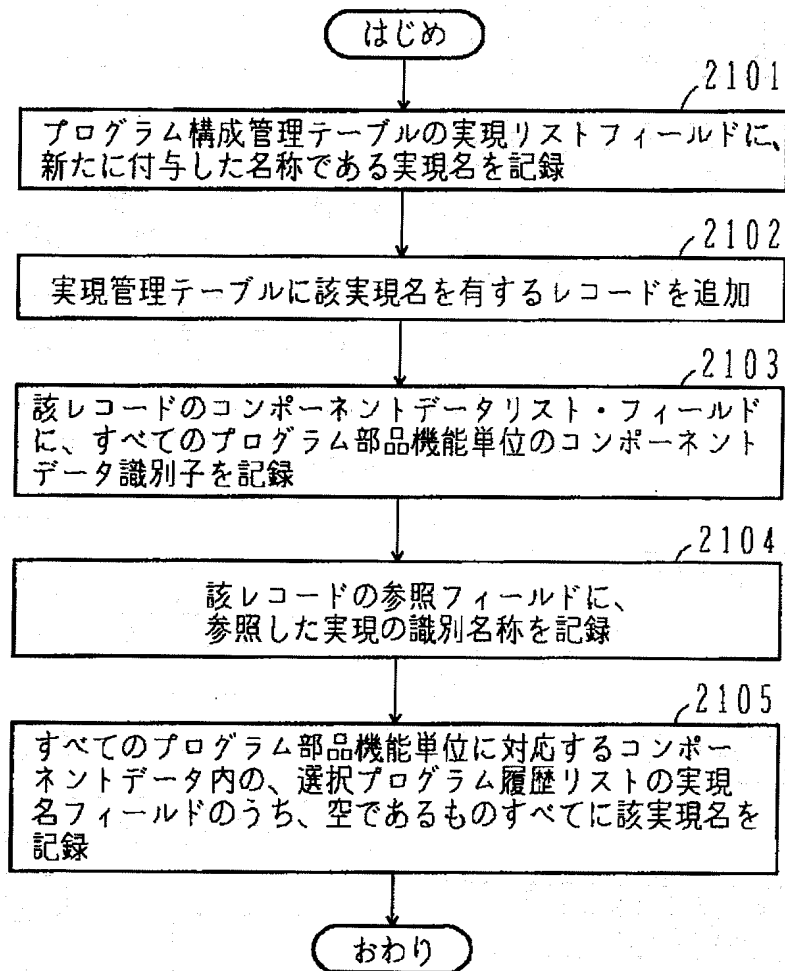
900

【図20】

206



【図21】

1511

【図25】

110

Program 1301 Components 1315 Candidates 1303

Classification	ID Name
在庫管理	*
受発注管理	OrderMgr
販売データ解	AnalysisMgr
在庫マスタ	StockDB
受発注マスタ	OrderDB
解析マスタ	AnalysisDB

1306 1307 1308 1309 1310 1311 1312

1320 1321

Run Stop

Message Log 1322 1323 1324

Message	From	Response
void Locked(void)	StockDB	StockMgr::OK
void Released(void)	StockDB	StockMgr::OK
void Add(Item, Count)	StockClient	StockMgr::OK
void Retrieve(Item, Count)	StockClient	StockMgr::OK
void Refer(Item, Count&, Date&, arrival)	StockClient	StockMgr::OK

1325

Example: 1332 1330 Register Remove 1331

Record 1340 1341 Exit

【図26】

110

Program 1301 Components 1315 Candidates 1303

Classification	ID Name
在庫管理	StockMgr
受発注管理	OrderMgr
販売データ解	AnalysisMgr
在庫マスタ	StockDB
受発注マスタ	OrderDB
解析マスタ	AnalysisDB

1306 1307 1308 1309 1310 1311 1312

1320 1321

Run Stop

Message Log 1322 1323 1324

Message	From	Response
void Locked(void)	StockDB	StockMgr::OK
void Released(void)	StockDB	StockMgr::OK
void Add(Item, Count)	StockClient	[Example]
void Retrieve(Item, Count)	StockClient	StockMgr::OK
void Refer(Item, Count&, Date&, arrival)	StockClient	StockMgr::OK

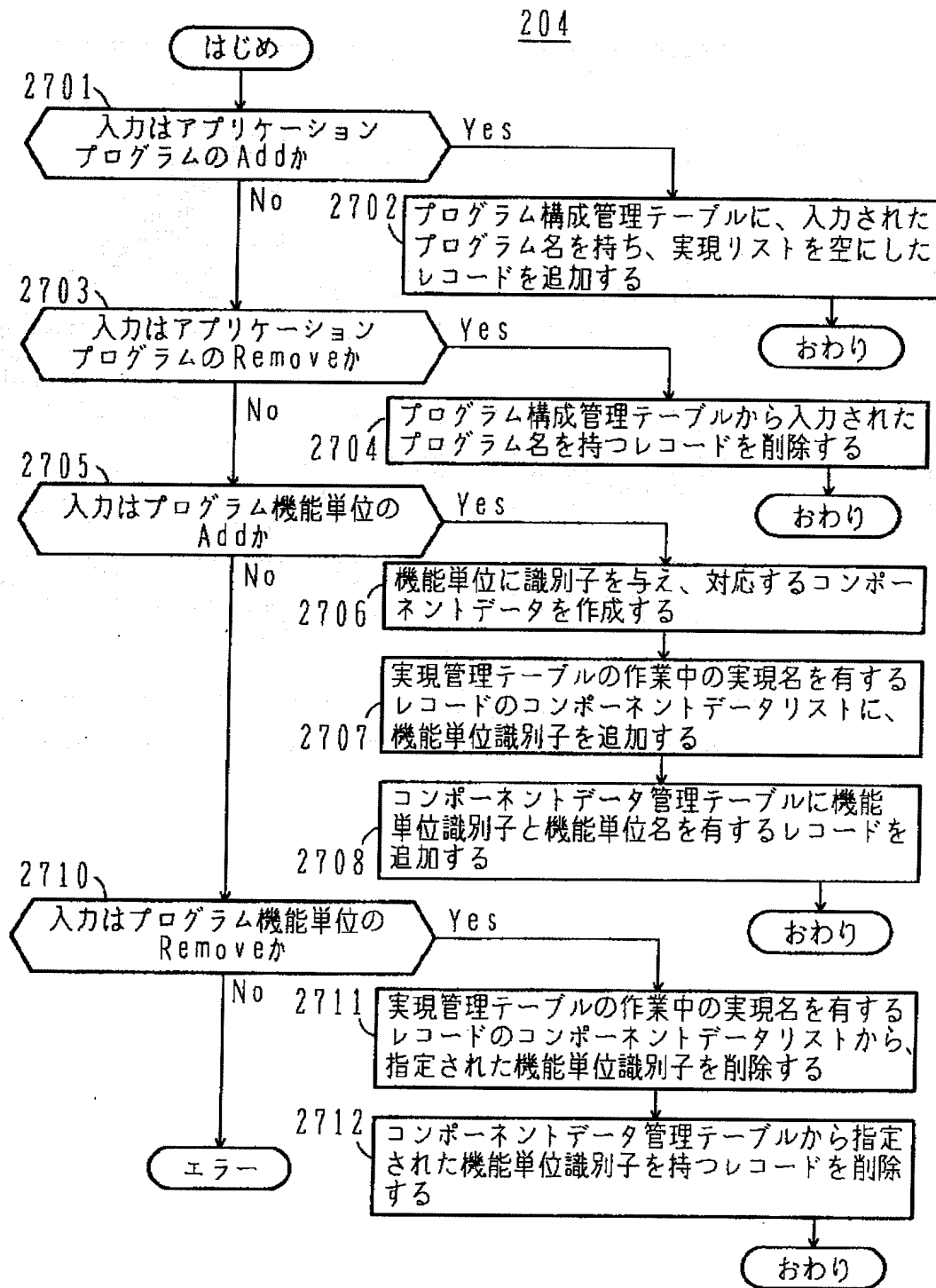
1325

Example: Add(Item item, Count count) 1332 Register Remove 1331

count<<"Item Added"<<item.Name<<count

Record 1340 1341 Exit

【図27】



2013

